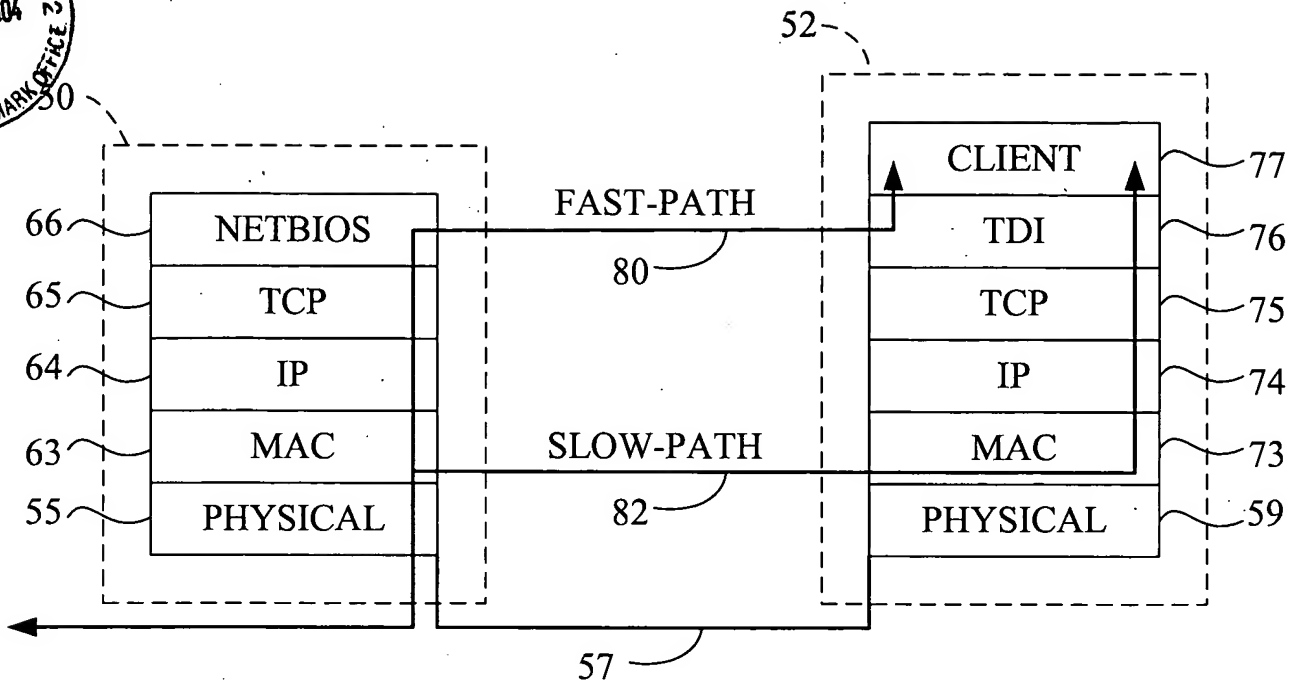

FIG. 1

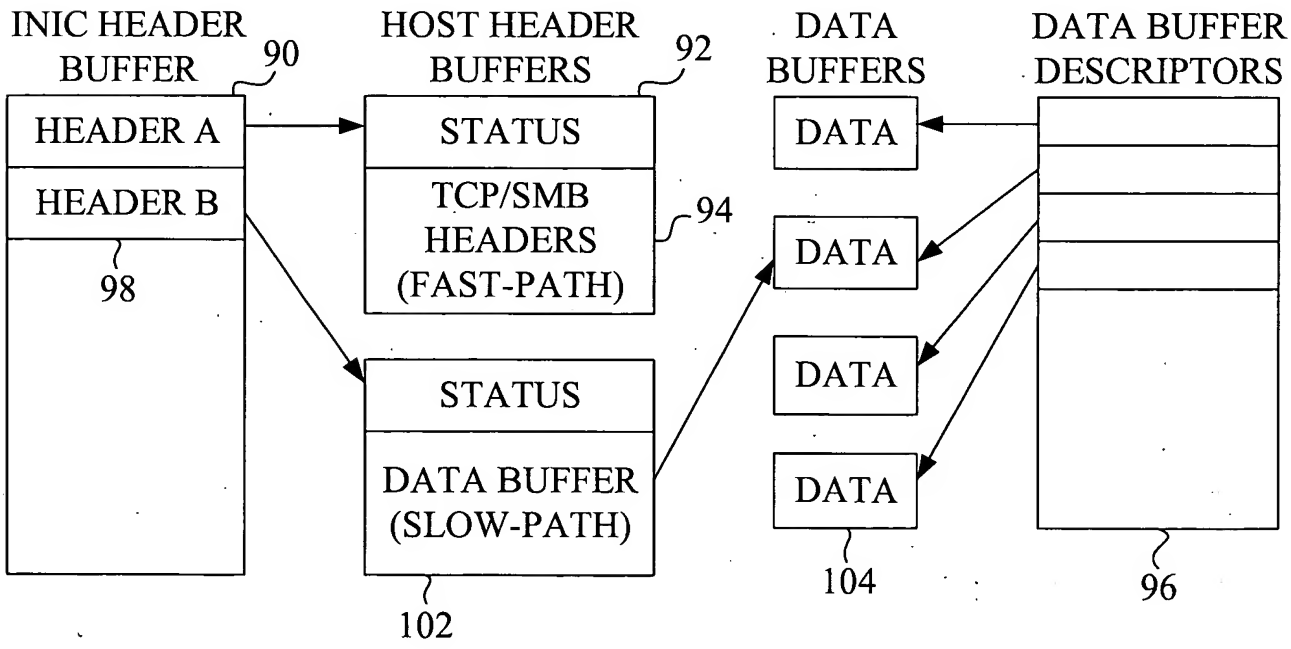

FIG. 2

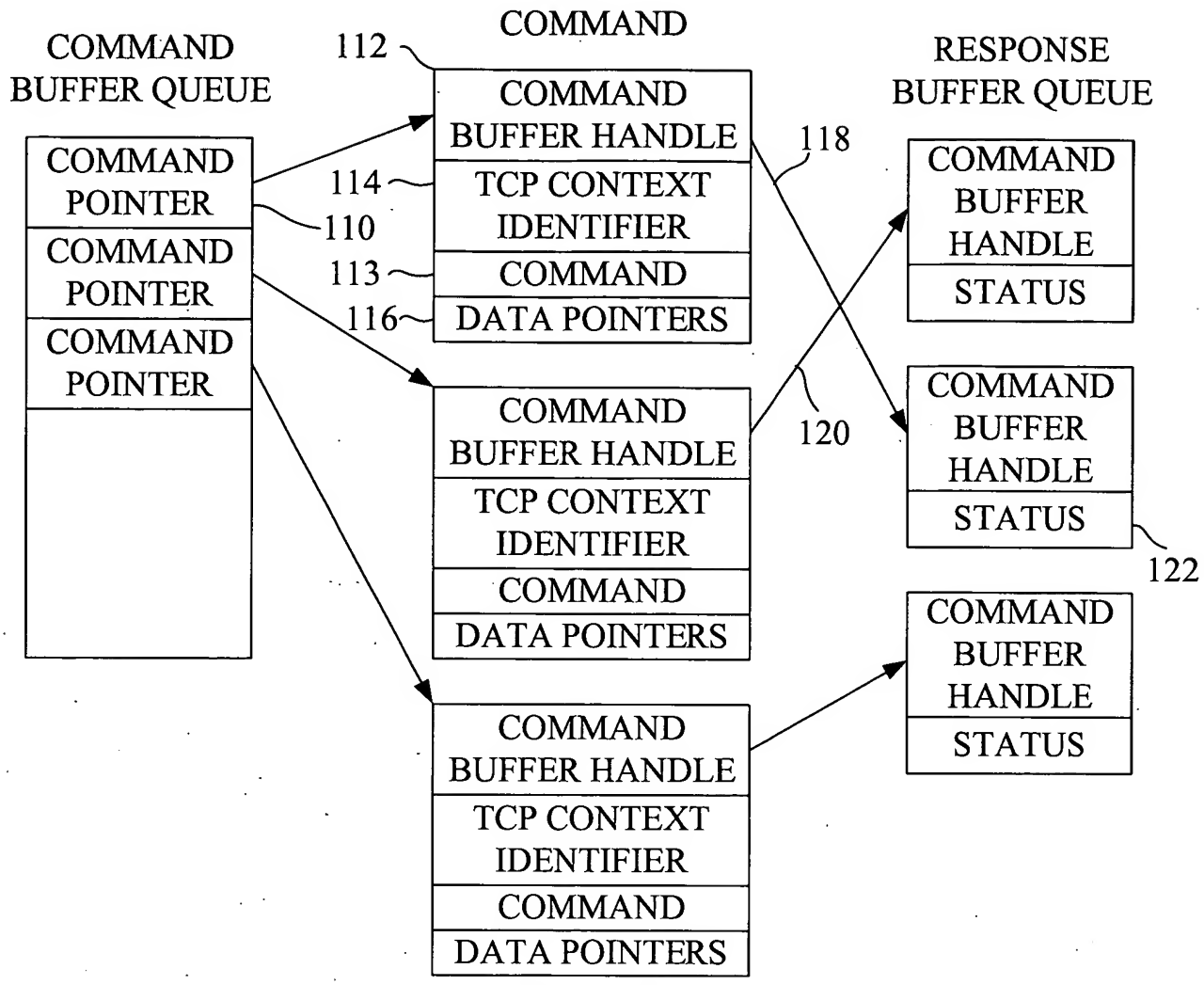
COMMAND BUFFER QUEUE
COMMAND POINTER
COMMAND POINTER
COMMAND POINTER
110
112
COMMAND
COMMAND BUFFER HANDLE
114
TCP CONTEXT IDENTIFIER
113
COMMAND
116
DATA POINTERS
COMMAND BUFFER HANDLE
TCP CONTEXT IDENTIFIER
COMMAND
DATA POINTERS
COMMAND BUFFER HANDLE
TCP CONTEXT IDENTIFIER
COMMAND
DATA POINTERS
118
120
RESPONSE BUFFER QUEUE
COMMAND BUFFER HANDLE
STATUS
COMMAND BUFFER HANDLE
STATUS
122
COMMAND BUFFER HANDLE
STATUS

FIG. 3

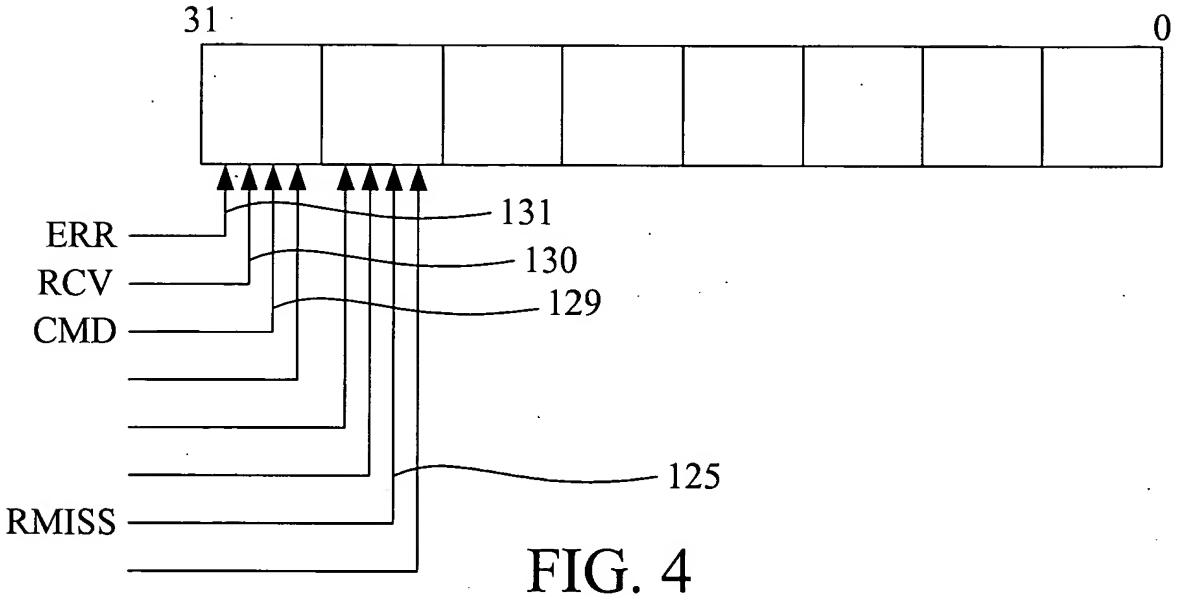
31
0
ERR
RCV
CMD
RMISS
131
130
129
125

FIG. 4

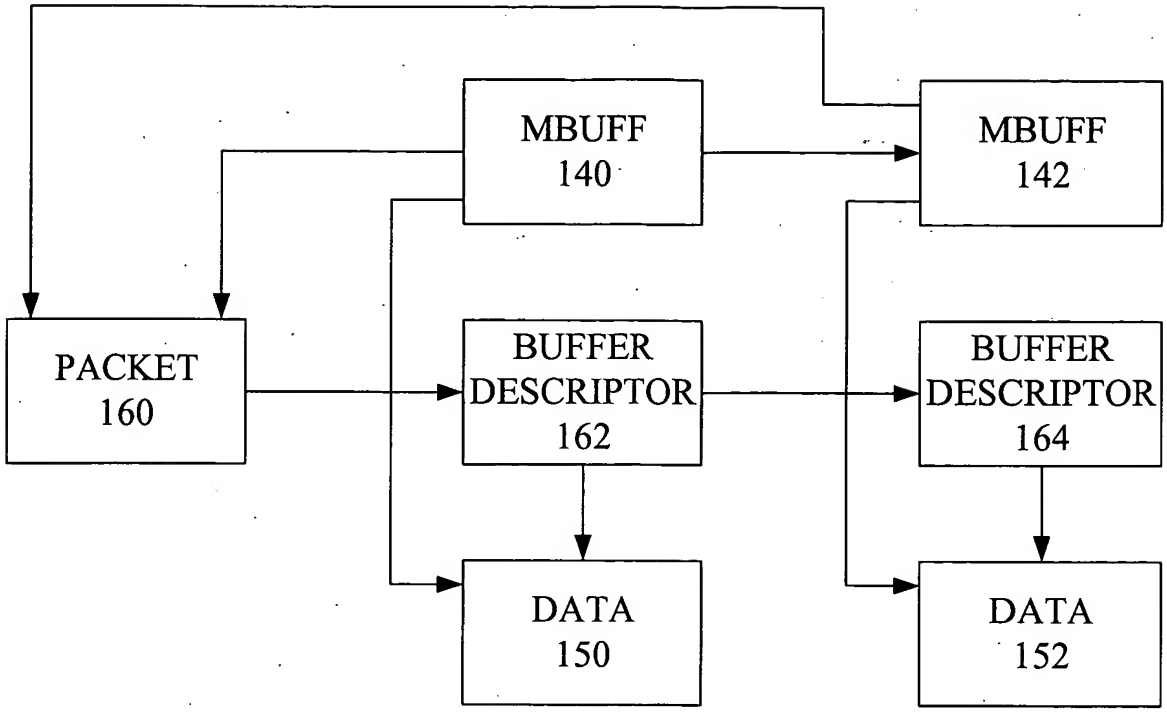
MBUFF
140
MBUFF
142
PACKET
160
BUFFER
DESCRIPTOR
162
BUFFER
DESCRIPTOR
164
DATA
150
DATA
152

FIG. 5

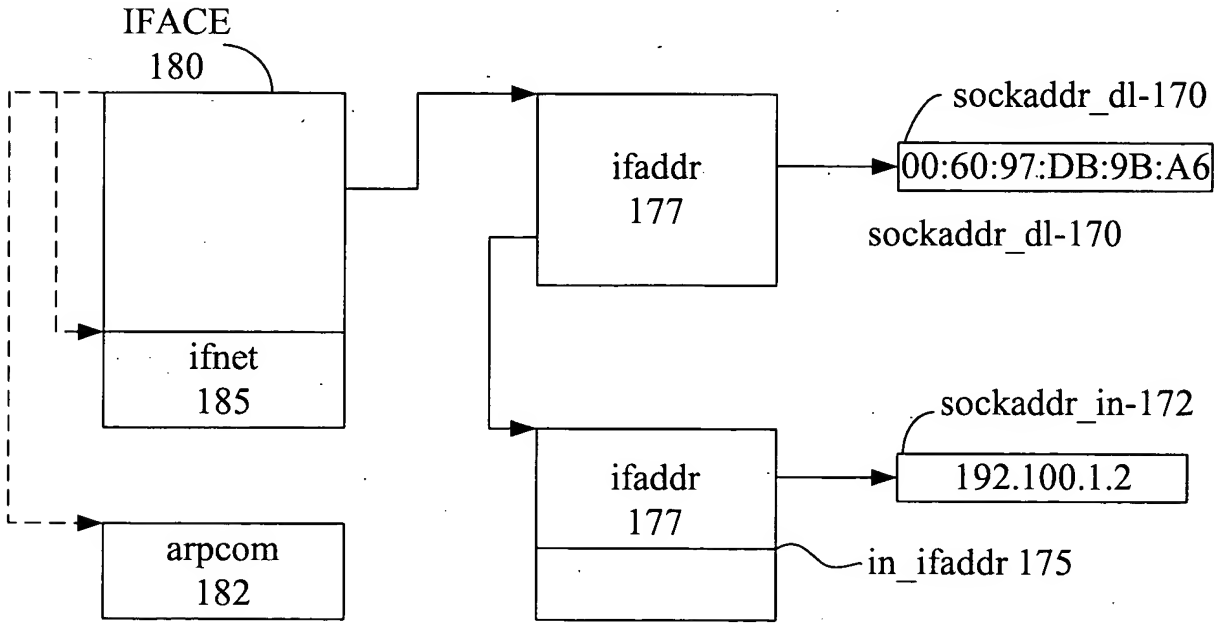
IFACE
180
ifnet
185
arpcom
182
ifaddr
177
sockaddr_dl-170
00:60:97:DB:9B:A6
sockaddr_dl-170
ifaddr
177
sockaddr_in-172
192.100.1.2
in_ifaddr 175

FIG. 6

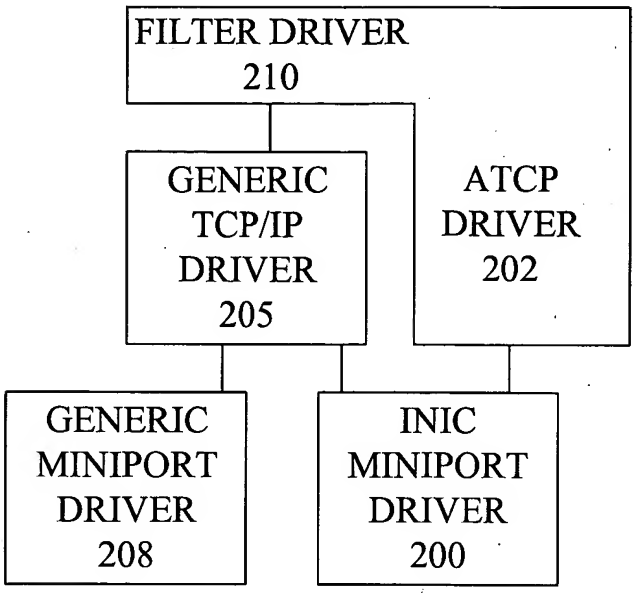

FIG. 7

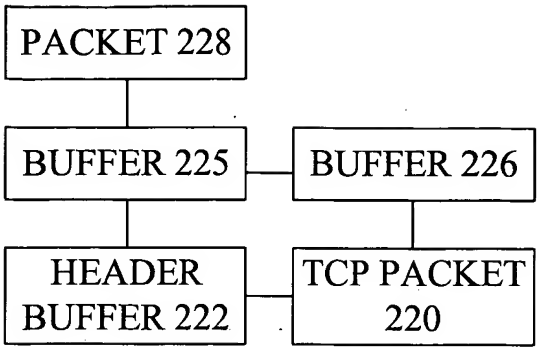

FIG. 8A

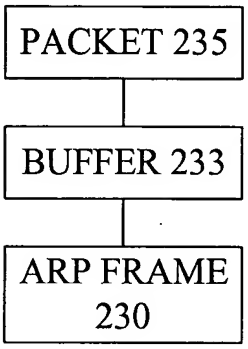

FIG. 8B

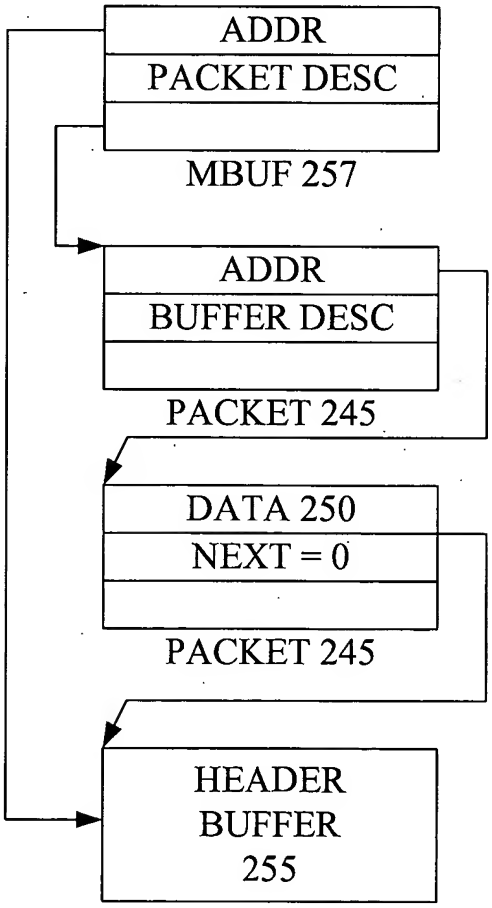

FIG. 9A

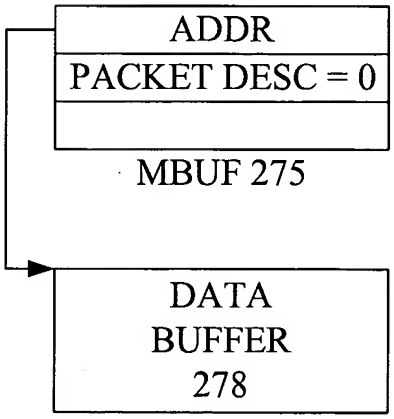

FIG. 9C

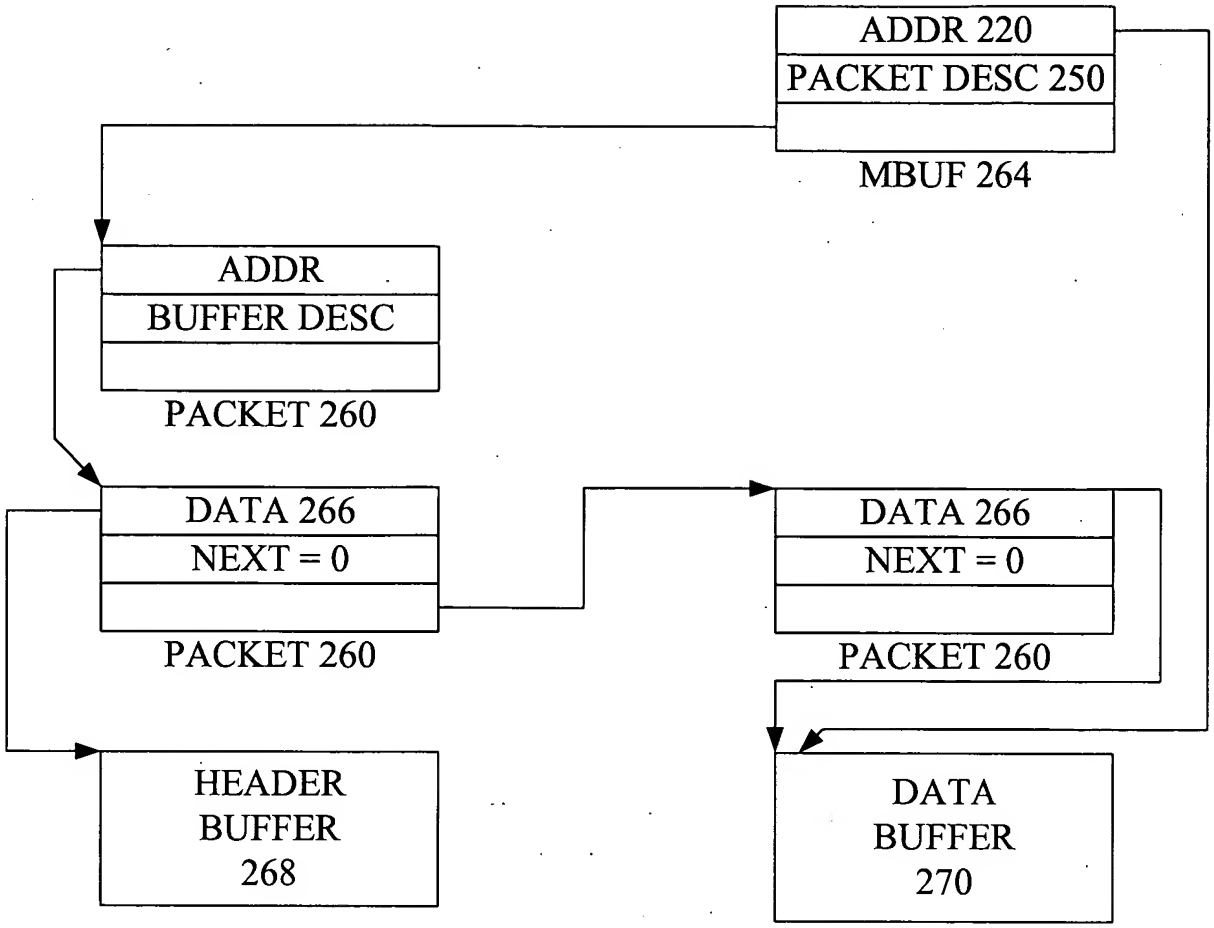
ADDR 220
PACKET DESC 250
MBUF 264
ADDR
BUFFER DESC
PACKET 260
DATA 266
NEXT = 0
PACKET 260
DATA 266
NEXT = 0
PACKET 260
HEADER
BUFFER
268
DATA
BUFFER
270

FIG. 9B

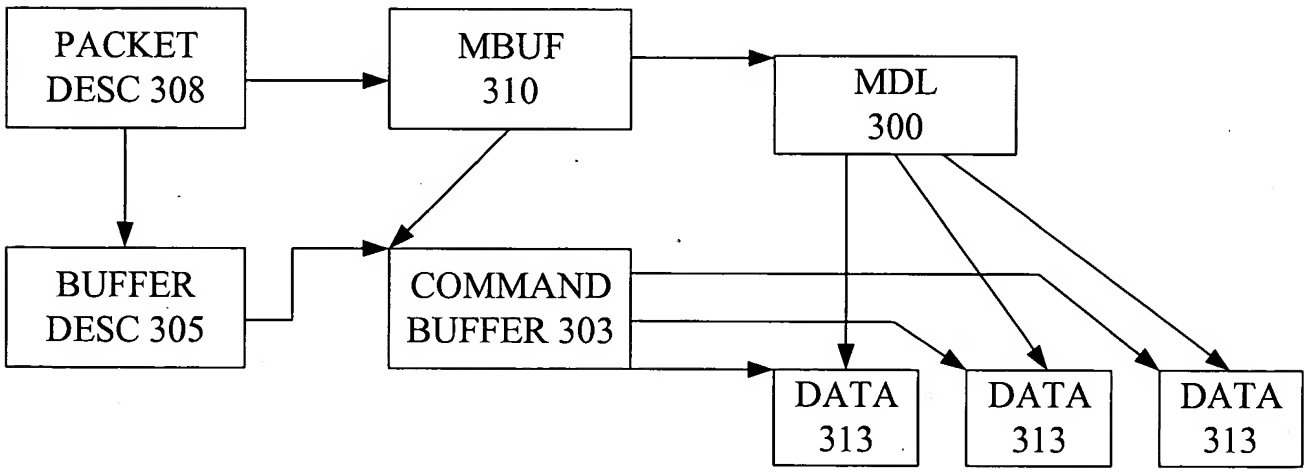
PACKET
DESC 308
MBUF
310
MDL
300
BUFFER
DESC 305
COMMAND
BUFFER 303
DATA
313
DATA
313
DATA
313

FIG. 10

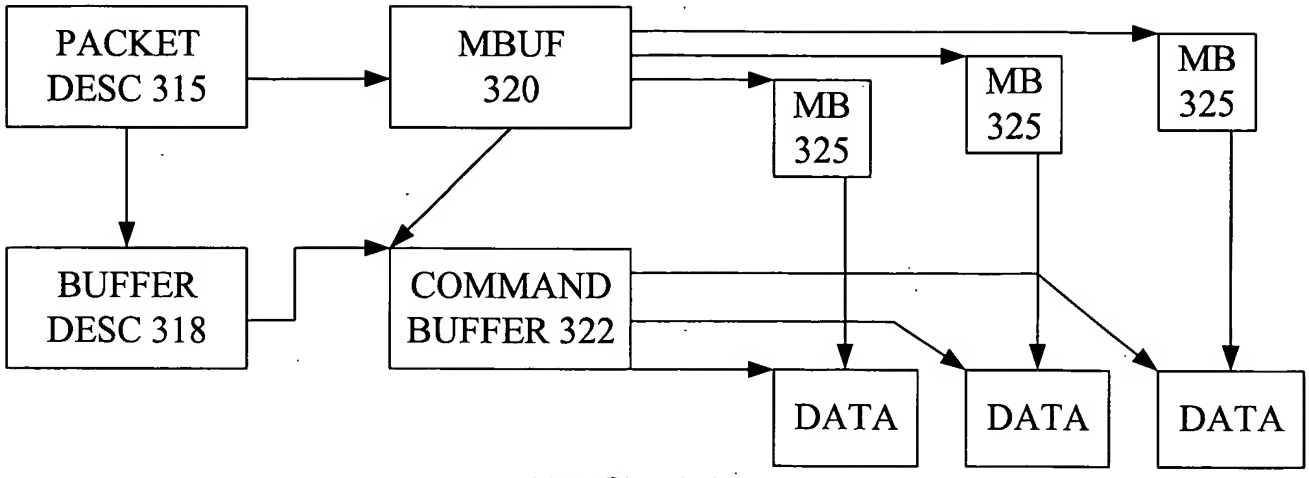

FIG. 11

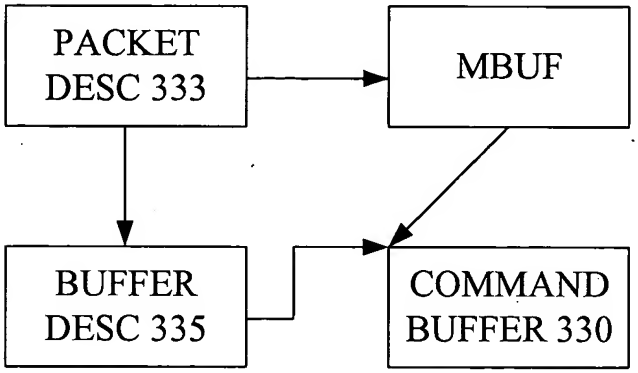

FIG. 12

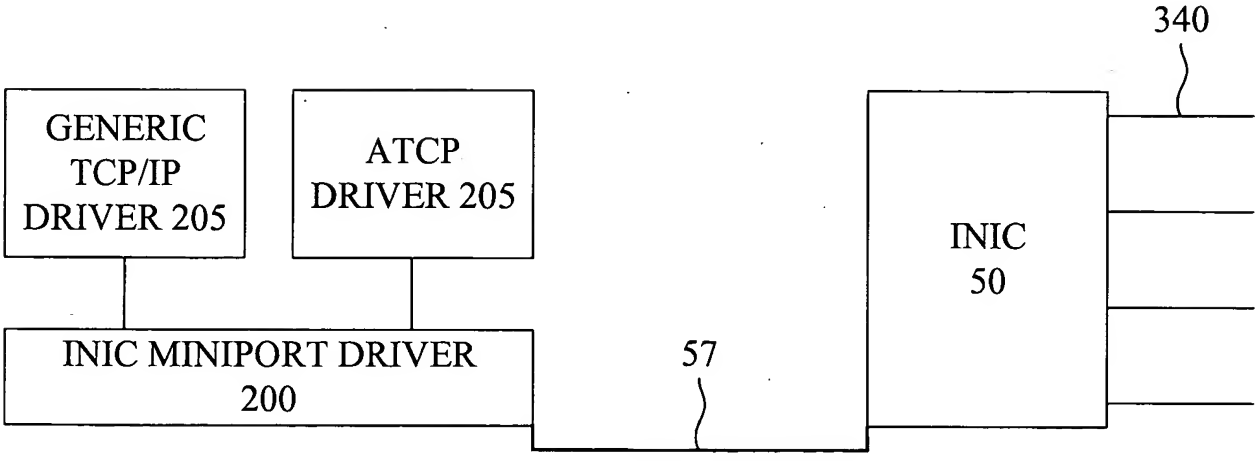

FIG. 13

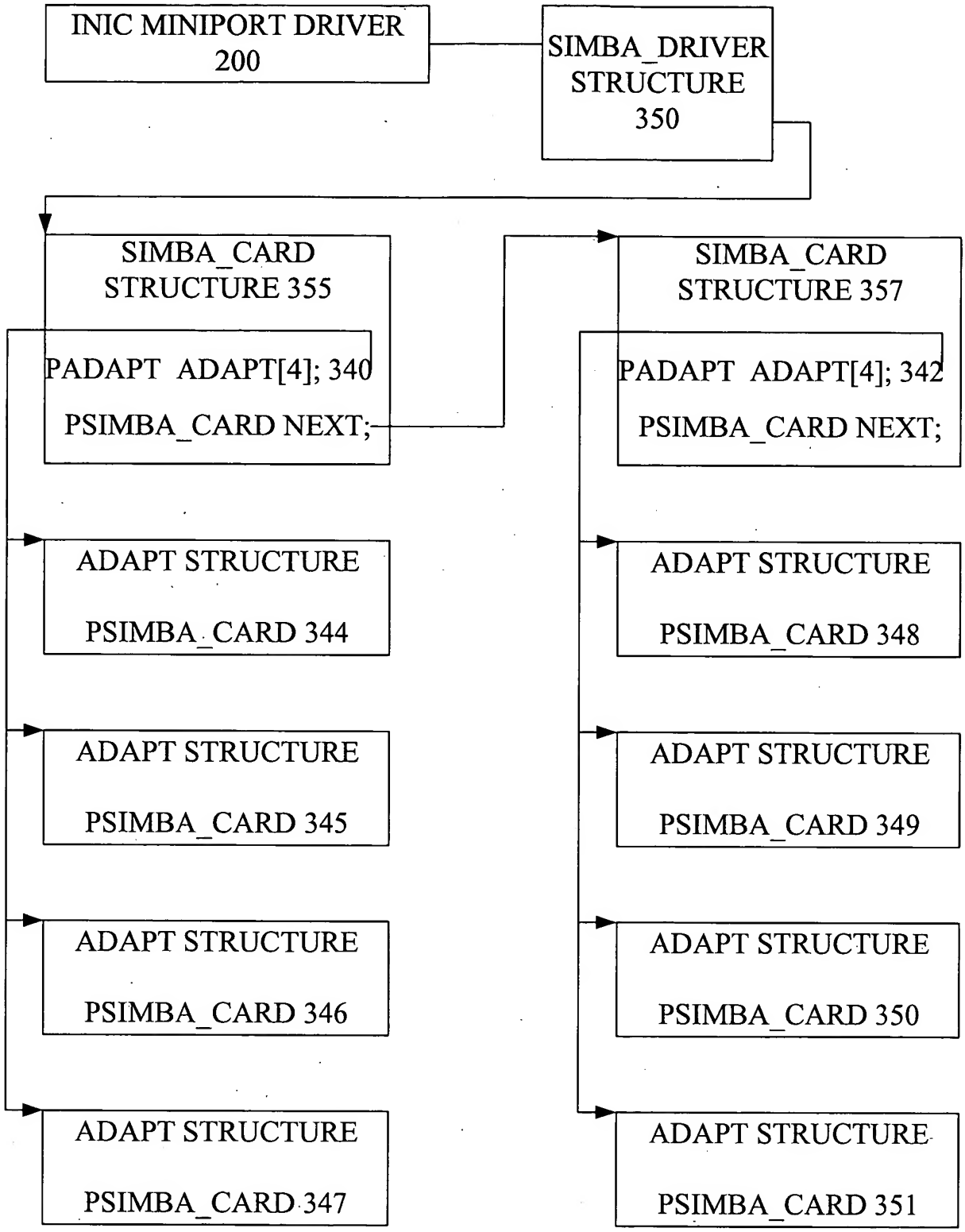
INIC MINIPORT DRIVER 200
SIMBA_DRIVER STRUCTURE 350
SIMBA_CARD STRUCTURE 355
PADAPT ADAPT[4]; 340
PSIMBA_CARD NEXT;
SIMBA_CARD STRUCTURE 357
PADAPT ADAPT[4]; 342
PSIMBA_CARD NEXT;
ADAPT STRUCTURE PSIMBA_CARD 344
ADAPT STRUCTURE PSIMBA_CARD 345
ADAPT STRUCTURE PSIMBA_CARD 346
ADAPT STRUCTURE PSIMBA_CARD 347
ADAPT STRUCTURE PSIMBA_CARD 348
ADAPT STRUCTURE PSIMBA_CARD 349
ADAPT STRUCTURE PSIMBA_CARD 350
ADAPT STRUCTURE PSIMBA_CARD 351

FIG. 14

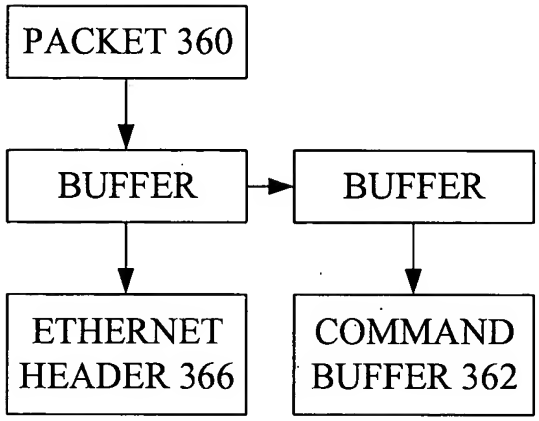

FIG. 15

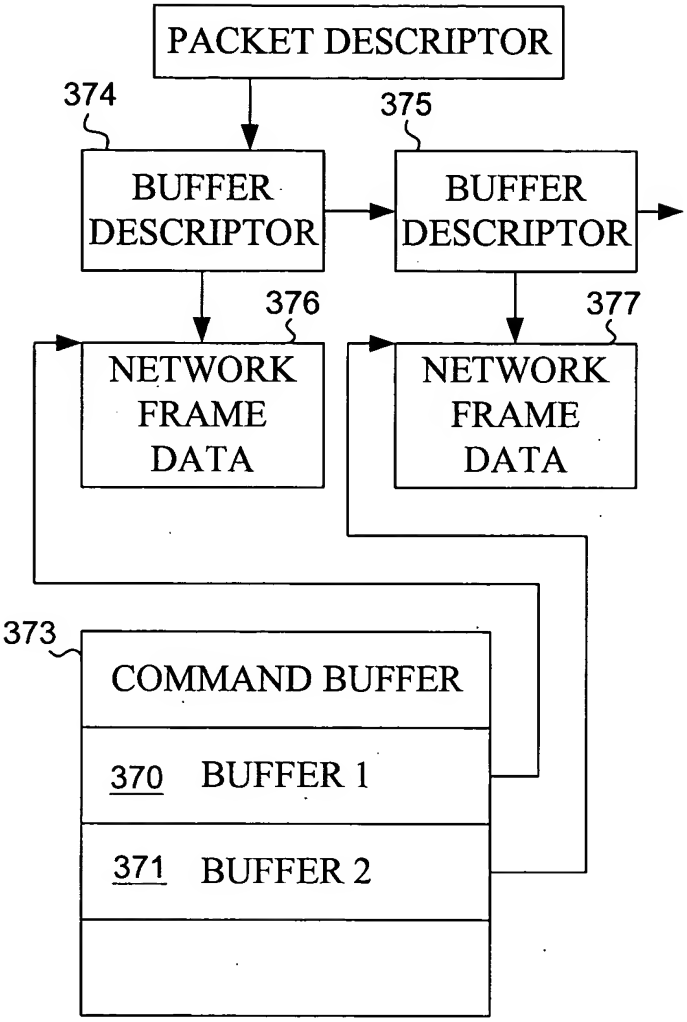

FIG. 16

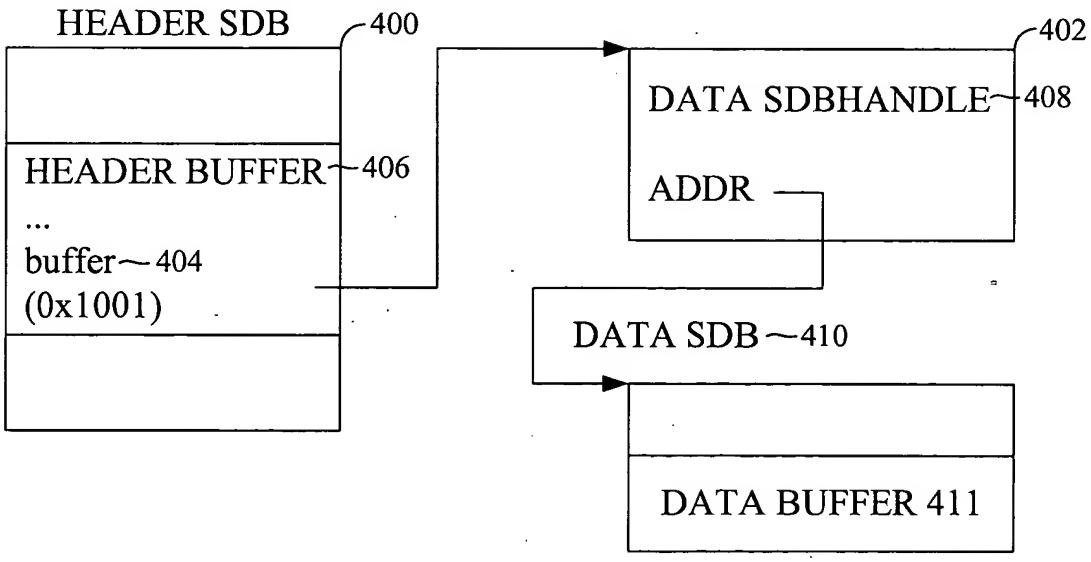

FIG. 17

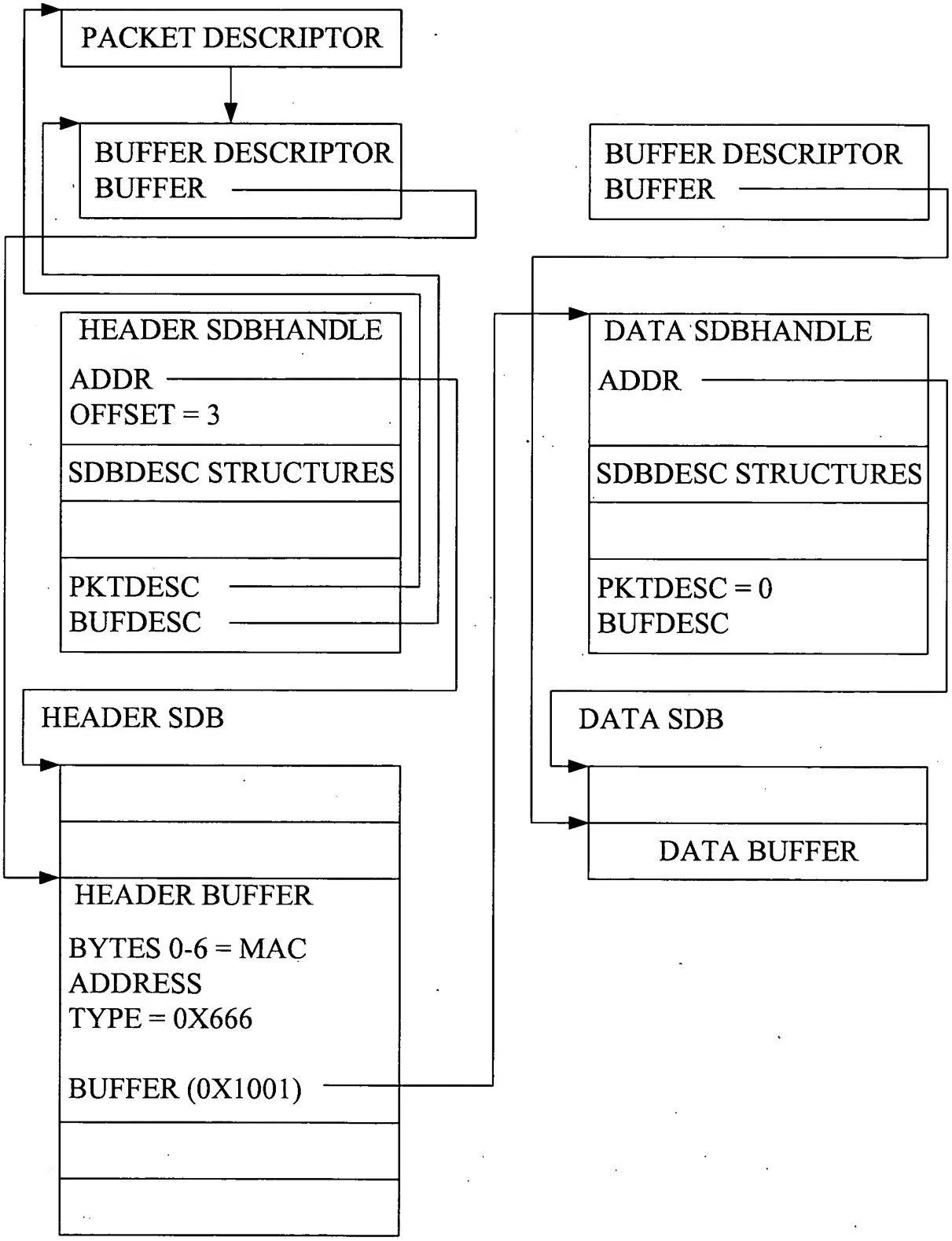
PACKET DESCRIPTOR
BUFFER DESCRIPTOR
BUFFER
BUFFER DESCRIPTOR
BUFFER
HEADER SDBHANDLE
ADDR
OFFSET = 3
SDBDESC STRUCTURES
PKTDESC
BUFDESC
DATA SDBHANDLE
ADDR
SDBDESC STRUCTURES
PKTDESC = 0
BUFDESC
HEADER SDB
DATA SDB
HEADER BUFFER
BYTES 0-6 = MAC ADDRESS
TYPE = 0X666
BUFFER (0X1001)
DATA BUFFER

FIG. 18

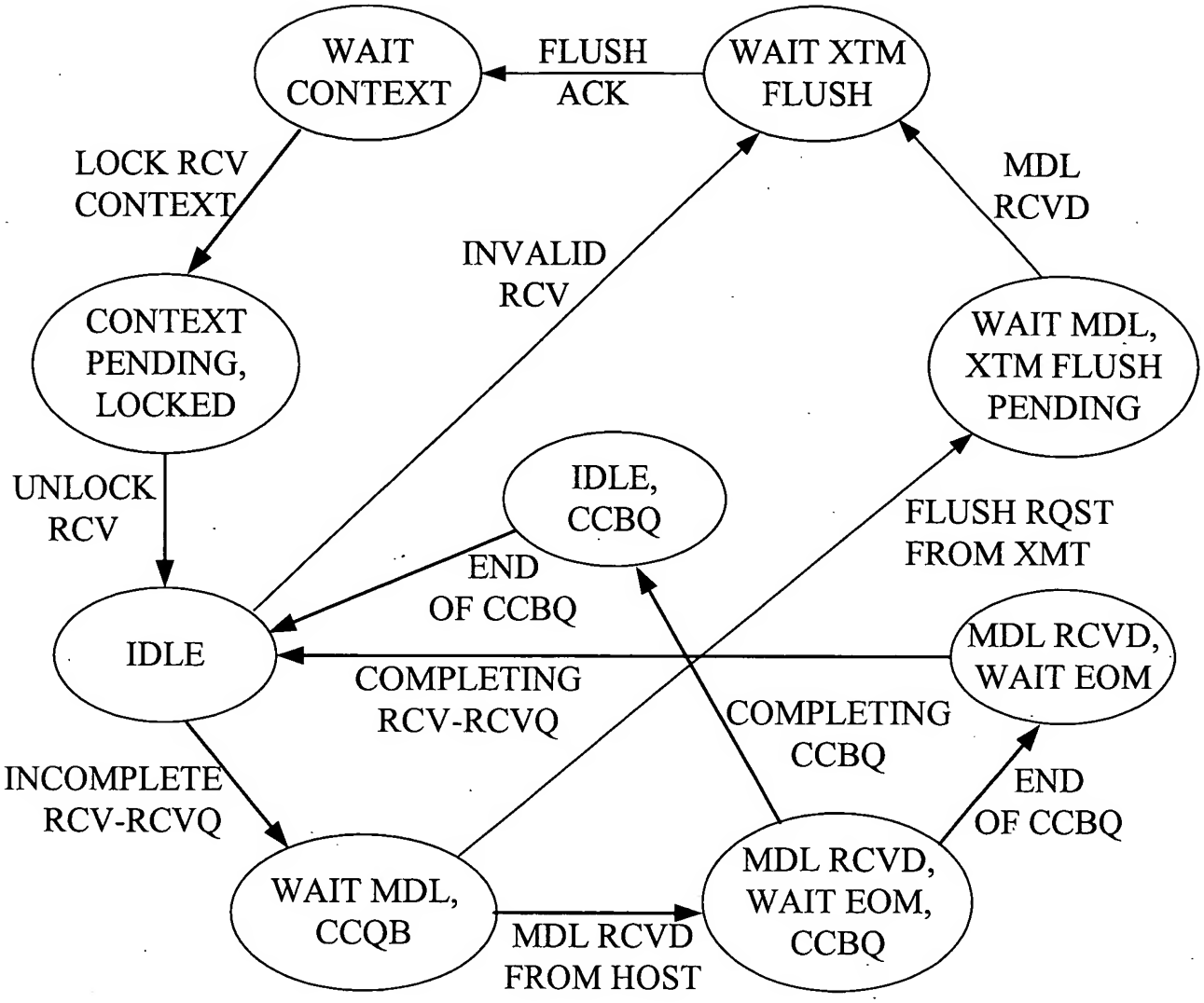
WAIT CONTEXT
FLUSH ACK
WAIT XTM FLUSH
LOCK RCV CONTEXT
MDL RCVD
INVALID RCV
CONTEXT PENDING, LOCKED
WAIT MDL, XTM FLUSH PENDING
UNLOCK RCV
IDLE, CCBQ
FLUSH RQST FROM XMT
END OF CCBQ
IDLE
MDL RCVD, WAIT EOM
COMPLETING RCV-RCVQ
COMPLETING CCBQ
INCOMPLETE RCV-RCVQ
END OF CCBQ
WAIT MDL, CCQB
MDL RCVD, WAIT EOM, CCBQ
MDL RCVD FROM HOST

FIG. 19

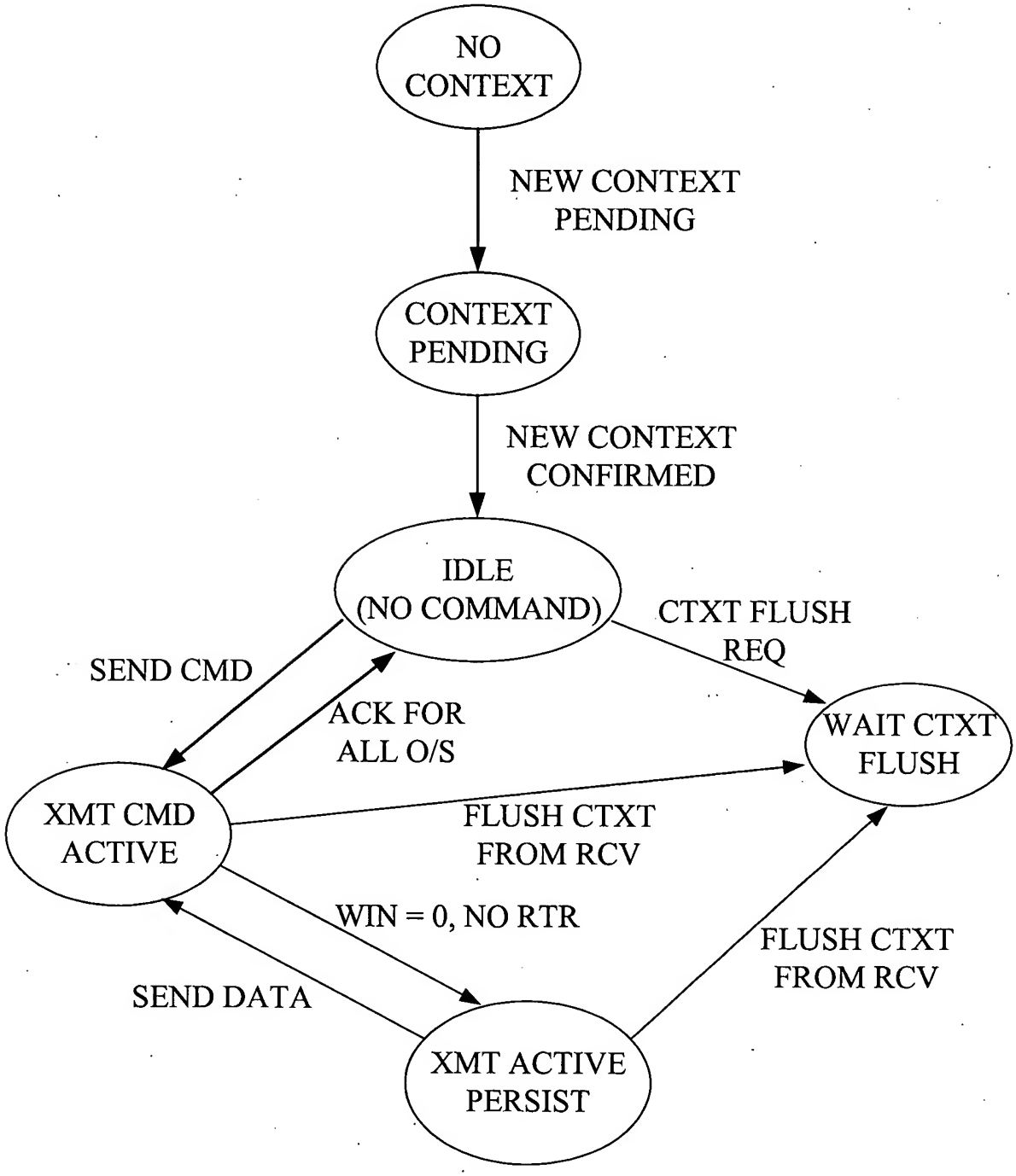
NO CONTEXT
NEW CONTEXT PENDING
CONTEXT PENDING
NEW CONTEXT CONFIRMED
IDLE (NO COMMAND)
CTXT FLUSH REQ
SEND CMD
ACK FOR ALL O/S
WAIT CTXT FLUSH
XMT CMD ACTIVE
FLUSH CTXT FROM RCV
WIN = 0, NO RTR
FLUSH CTXT FROM RCV
SEND DATA
XMT ACTIVE PERSIST

FIG. 20

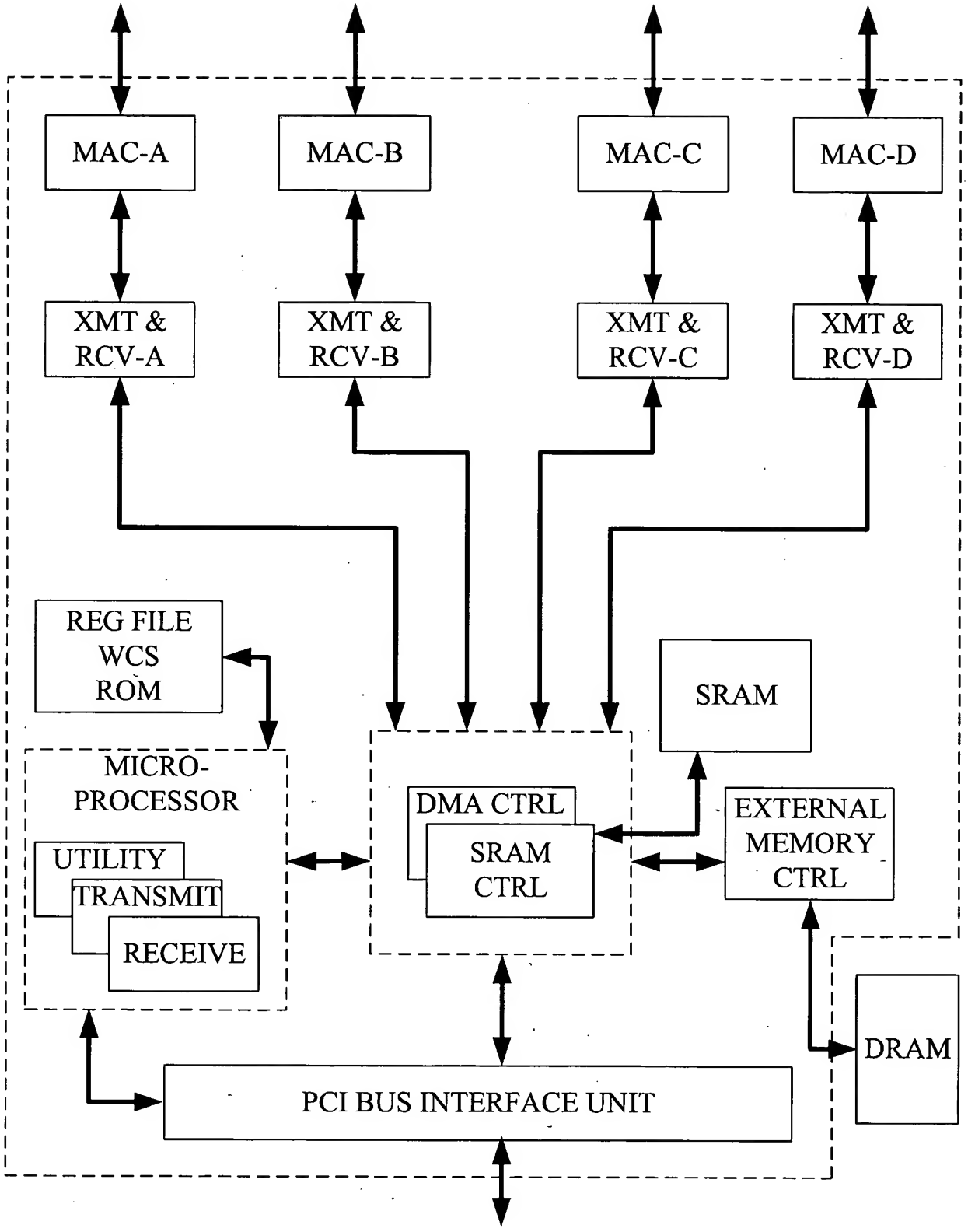
MAC-A
MAC-B
MAC-C
MAC-D
XMT & RCV-A
XMT & RCV-B
XMT & RCV-C
XMT & RCV-D
REG FILE WCS ROM
MICRO-PROCESSOR
UTILITY
TRANSMIT
RECEIVE
DMA CTRL
SRAM CTRL
SRAM
EXTERNAL MEMORY CTRL
DRAM
PCI BUS INTERFACE UNIT

FIG. 21

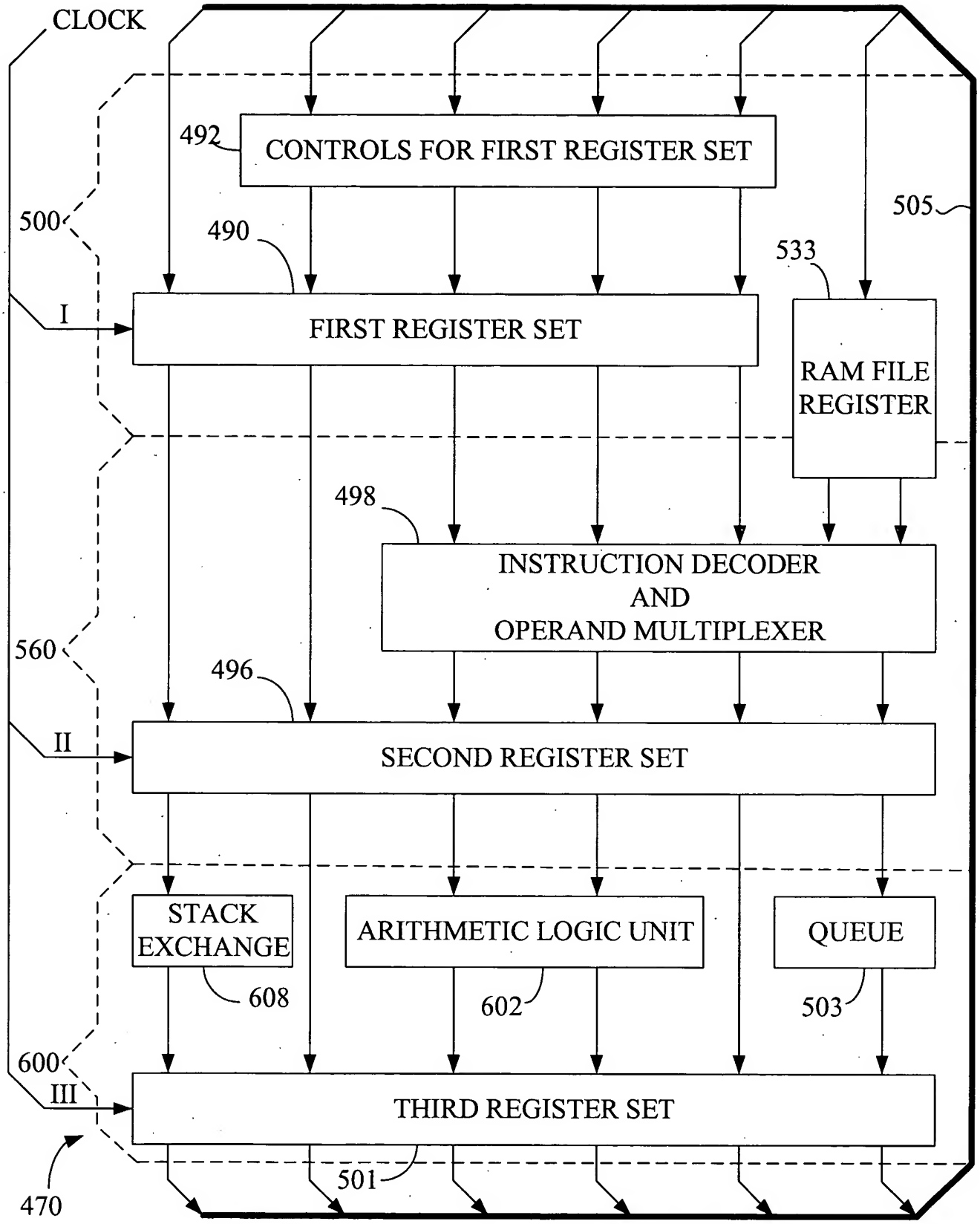
CLOCK
492
CONTROLS FOR FIRST REGISTER SET
505
500
490
533
I
FIRST REGISTER SET
RAM FILE REGISTER
498
INSTRUCTION DECODER AND OPERAND MULTIPLEXER
560
496
II
SECOND REGISTER SET
STACK EXCHANGE
608
ARITHMETIC LOGIC UNIT
602
QUEUE
503
600
III
THIRD REGISTER SET
501
470

FIG. 22

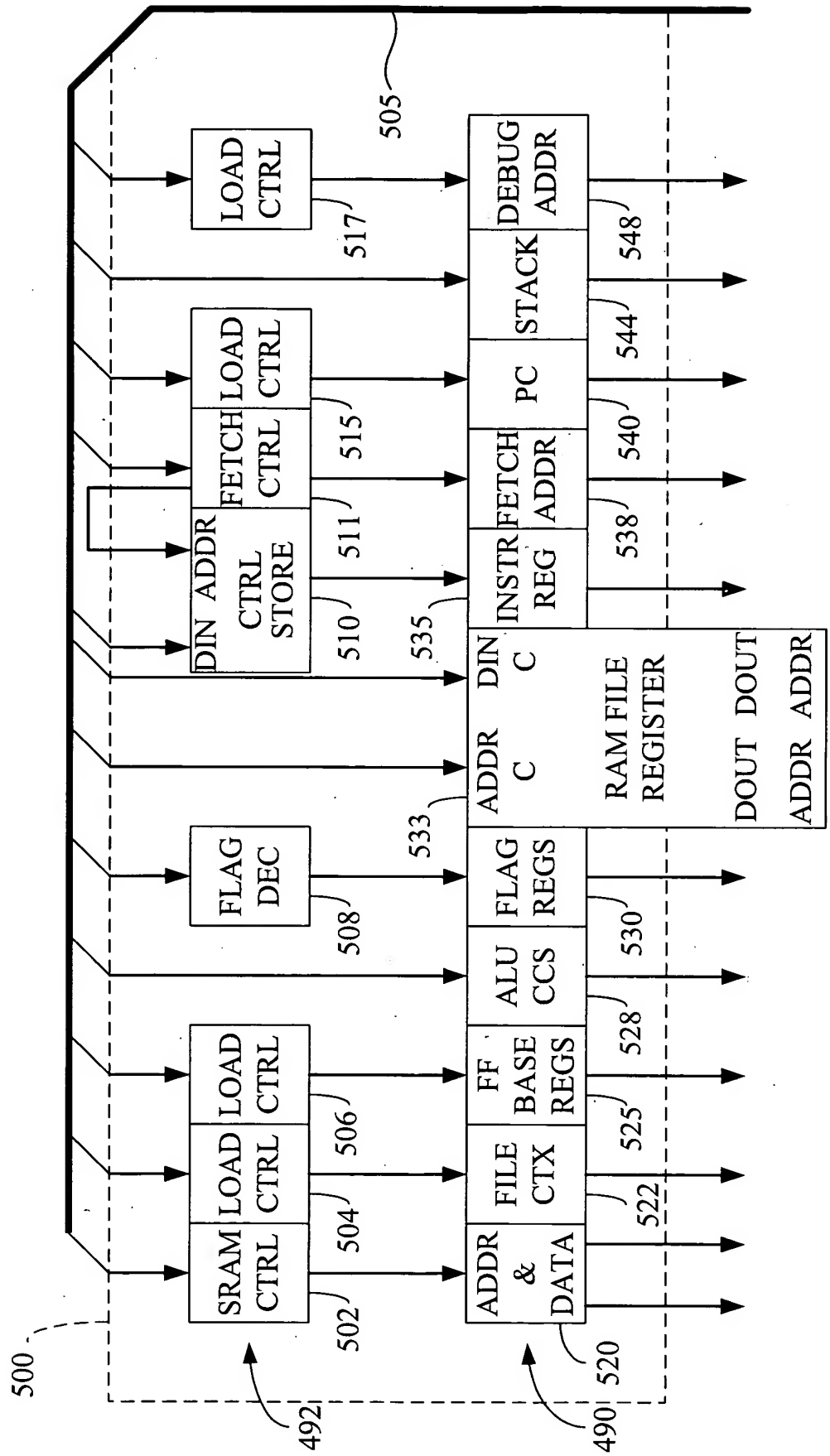
500
505
492
490
502
504
506
508
510
511
515
517
520
522
525
528
530
533
535
538
540
544
548
SRAM CTRL
LOAD CTRL
LOAD CTRL
FLAG DEC
DIN ADDR CTRL STORE
FETCH CTRL
LOAD CTRL
ADDR & DATA
FILE CTX
FF BASE REGS
ALU CCS
FLAG REGS
ADDR C
DIN C
INSTR REG
FETCH ADDR
PC
STACK
DEBUG ADDR
RAM FILE REGISTER
DOUT DOUT
ADDR ADDR

FIG. 23A

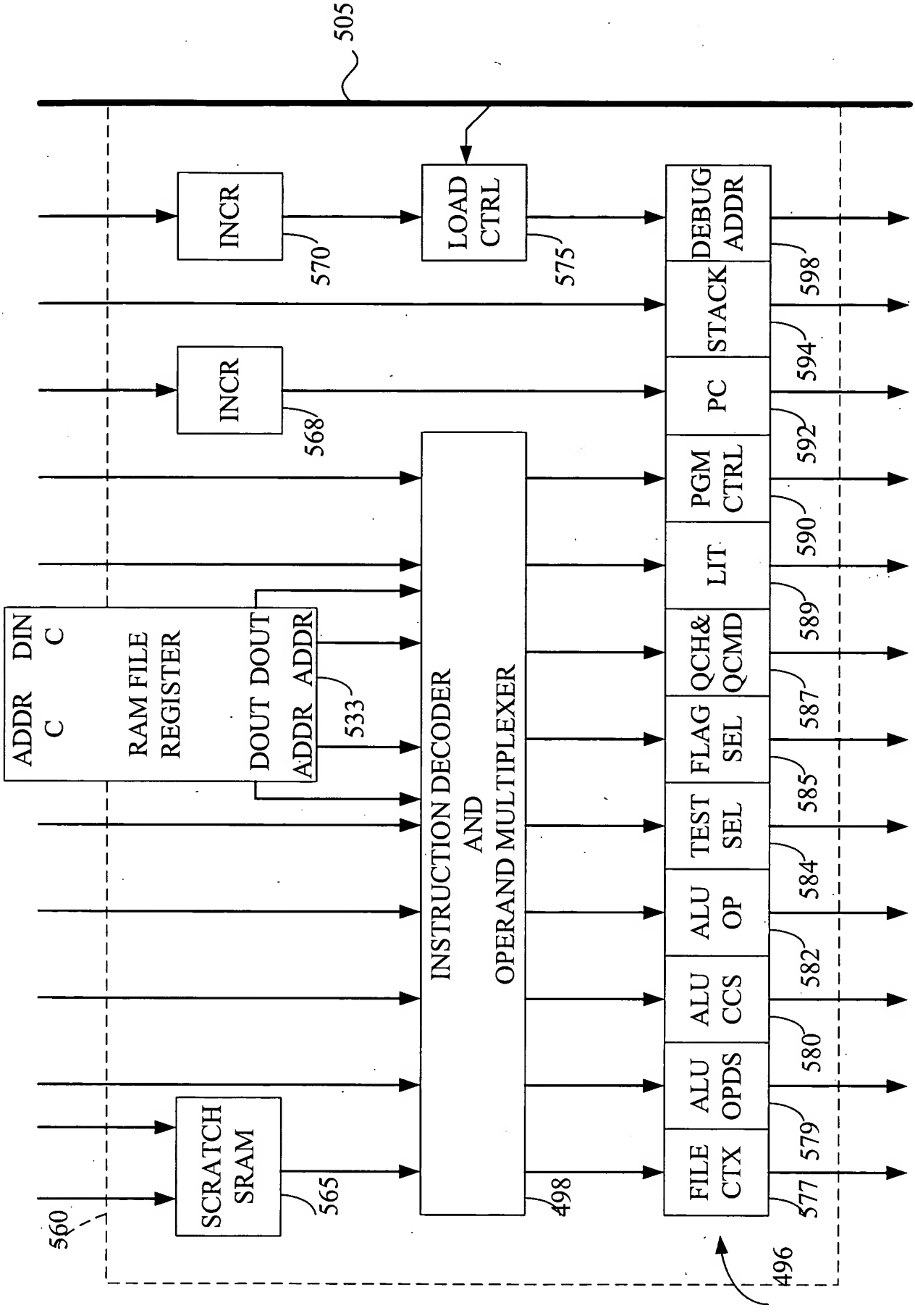
505
INCR
570
LOAD CTRL
575
DEBUG ADDR
598
INCR
568
STACK
594
PC
592
PGM CTRL
590
LIT
589
QCH& QCMD
587
FLAG SEL
585
TEST SEL
584
ALU OP
582
ALU CCS
580
ALU OPDS
579
FILE CTX
577
ADDR C
DIN C
RAM FILE REGISTER
DOUT DOUT
ADDR ADDR
533
INSTRUCTION DECODER AND OPERAND MULTIPLEXER
498
SCRATCH SRAM
565
560
496

FIG. 23B

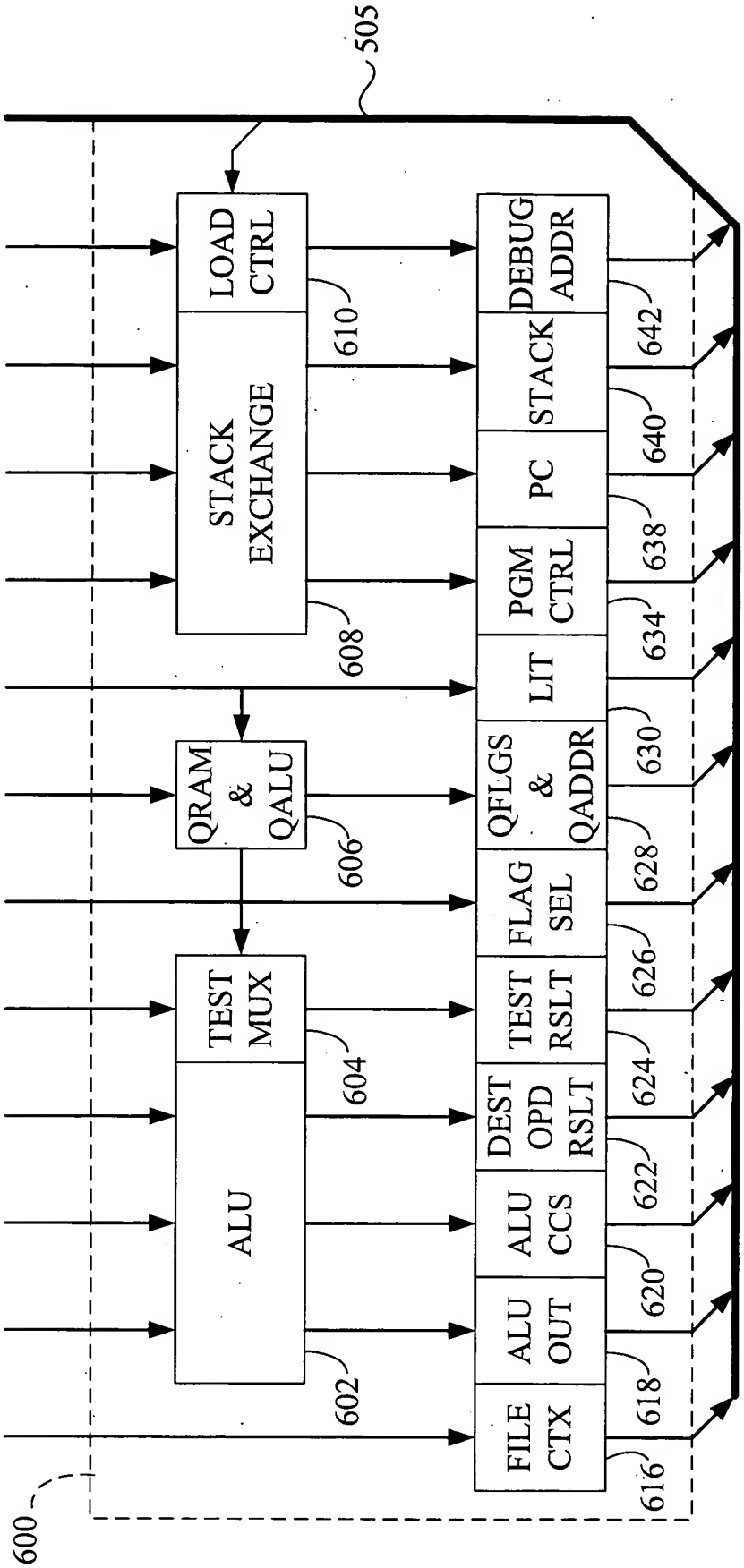
505
LOAD CTRL
STACK EXCHANGE
QRAM & QALU
TEST MUX
ALU
DEBUG ADDR
STACK
PC
PGM CTRL
LIT
QFLGS & QADDR
FLAG SEL
TEST RSLT
DEST OPD RSLT
ALU CCS
ALU OUT
FILE CTX
610
608
606
604
602
642
640
638
634
630
628
626
624
622
620
618
616
600

FIG. 23C

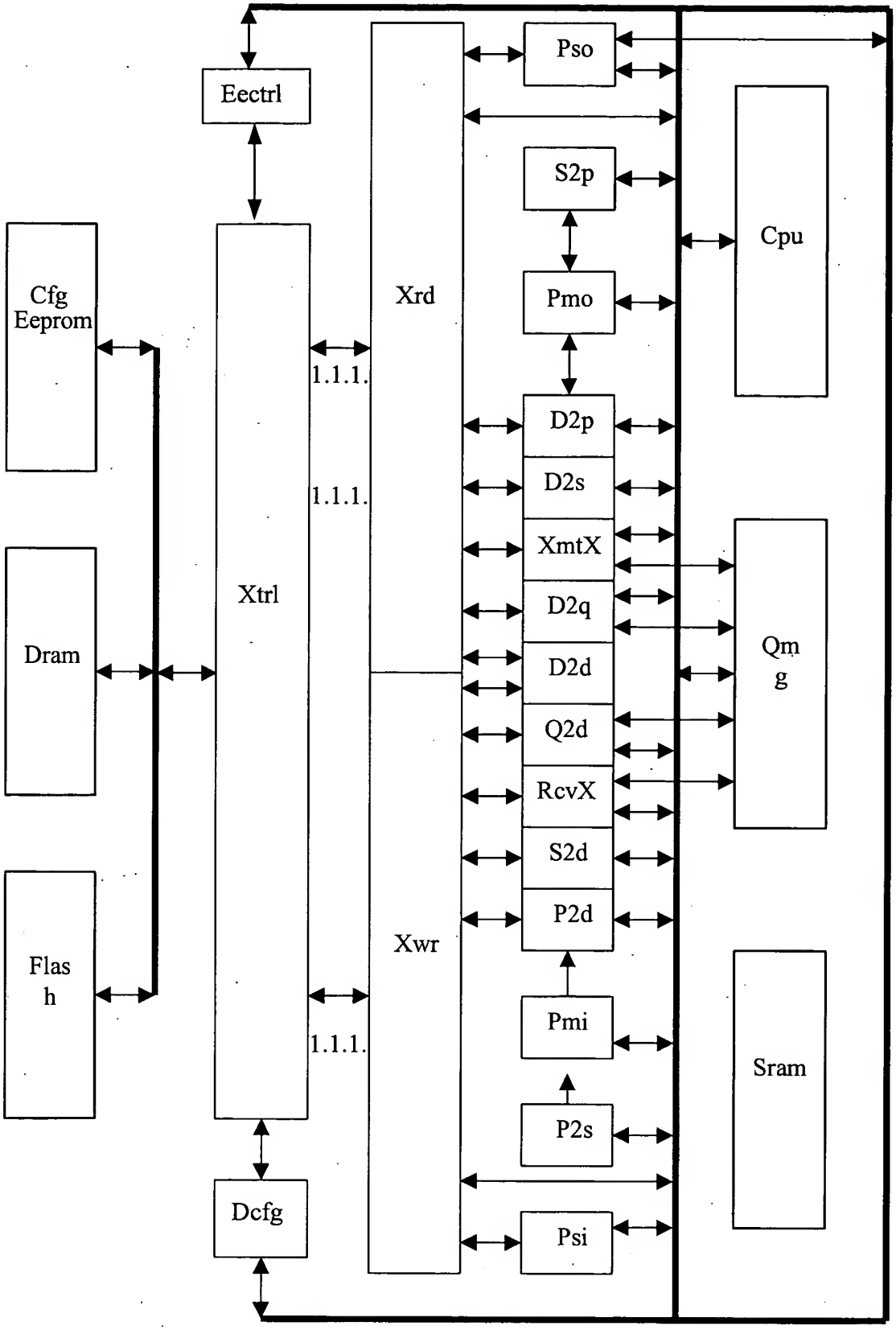
Eectrl
Cfg Eeprom
Dram
Flas h
Xtrl
Dcfg
1.1.1.
1.1.1.
1.1.1.
Xrd
Xwr
Pso
S2p
Pmo
D2p
D2s
XmtX
D2q
D2d
Q2d
RcvX
S2d
P2d
Pmi
P2s
Psi
Cpu
Qm g
Sram

FIG. 24

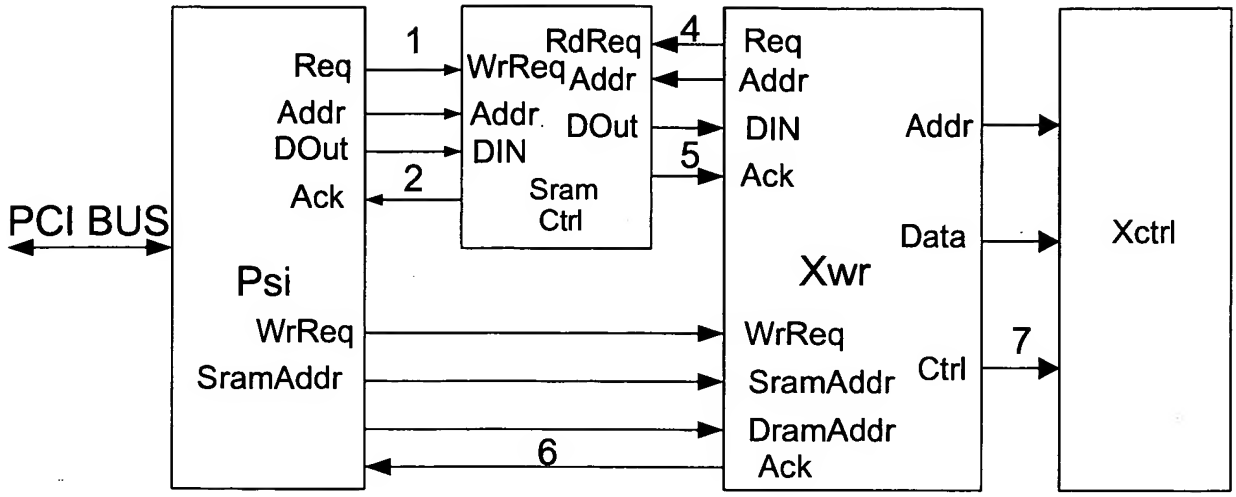

FIG. 25

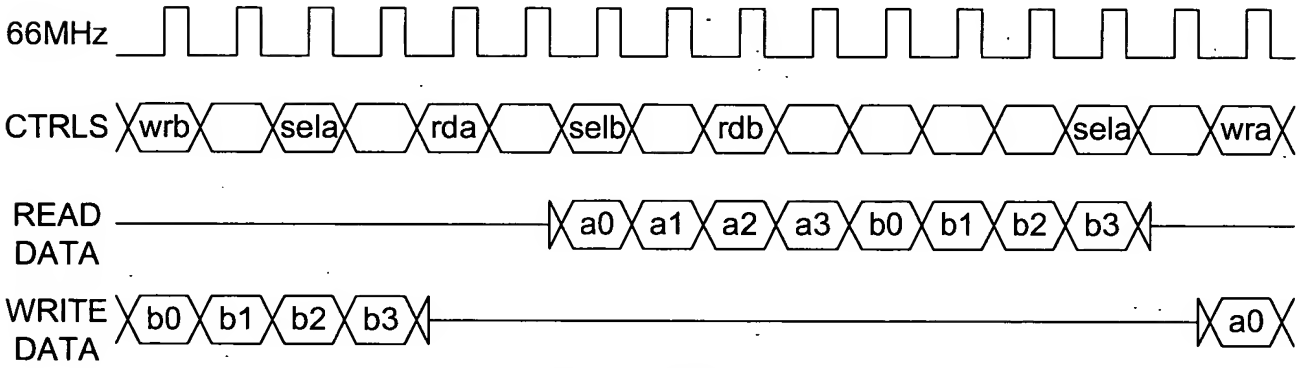

FIG. 29

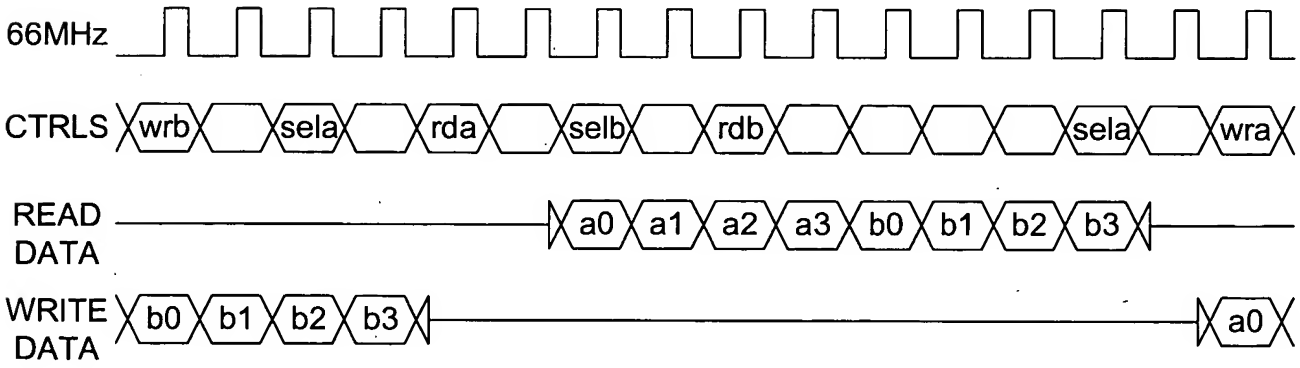

FIG. 31

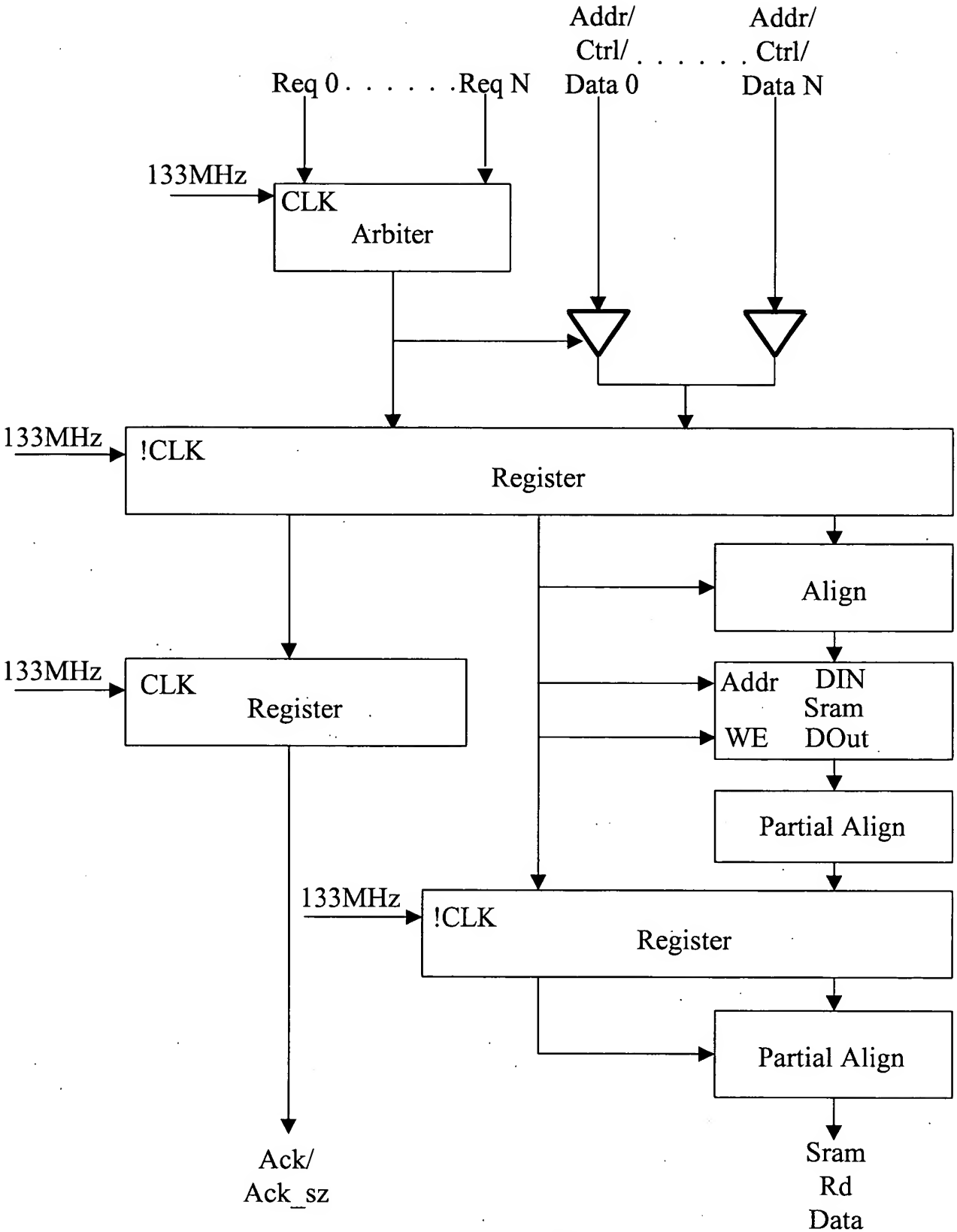
Addr/
Ctrl/
Data 0
Addr/
Ctrl/
Data N
Req 0 . . . . . Req N
133MHz
CLK
Arbiter
133MHz
!CLK
Register
Align
133MHz
CLK
Register
Addr
DIN
Sram
WE
DOut
Partial Align
133MHz
!CLK
Register
Partial Align
Ack/
Ack_sz
Sram
Rd
Data

FIG. 26

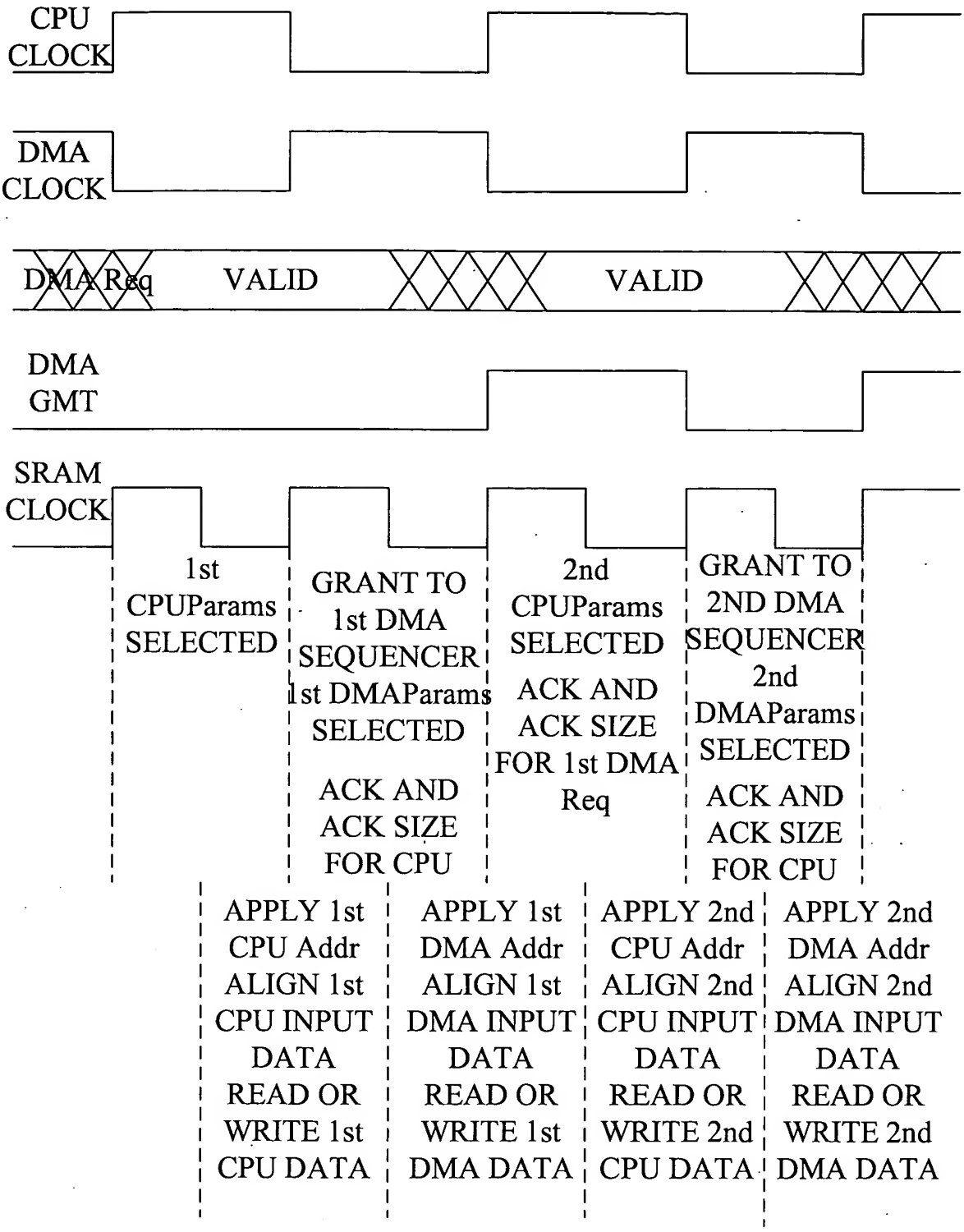
CPU CLOCK
DMA CLOCK
DMA Req
VALID
VALID
DMA GMT
SRAM CLOCK
1st CPUParams SELECTED
GRANT TO 1st DMA SEQUENCER 1st DMAParams SELECTED
ACK AND ACK SIZE FOR CPU
2nd CPUParams SELECTED
ACK AND ACK SIZE FOR 1st DMA Req
GRANT TO 2ND DMA SEQUENCER 2nd DMAParams SELECTED
ACK AND ACK SIZE FOR CPU
APPLY 1st CPU Addr ALIGN 1st CPU INPUT DATA READ OR WRITE 1st CPU DATA
APPLY 1st DMA Addr ALIGN 1st DMA INPUT DATA READ OR WRITE 1st DMA DATA
APPLY 2nd CPU Addr ALIGN 2nd CPU INPUT DATA READ OR WRITE 2nd CPU DATA
APPLY 2nd DMA Addr ALIGN 2nd DMA INPUT DATA READ OR WRITE 2nd DMA DATA

FIG. 27

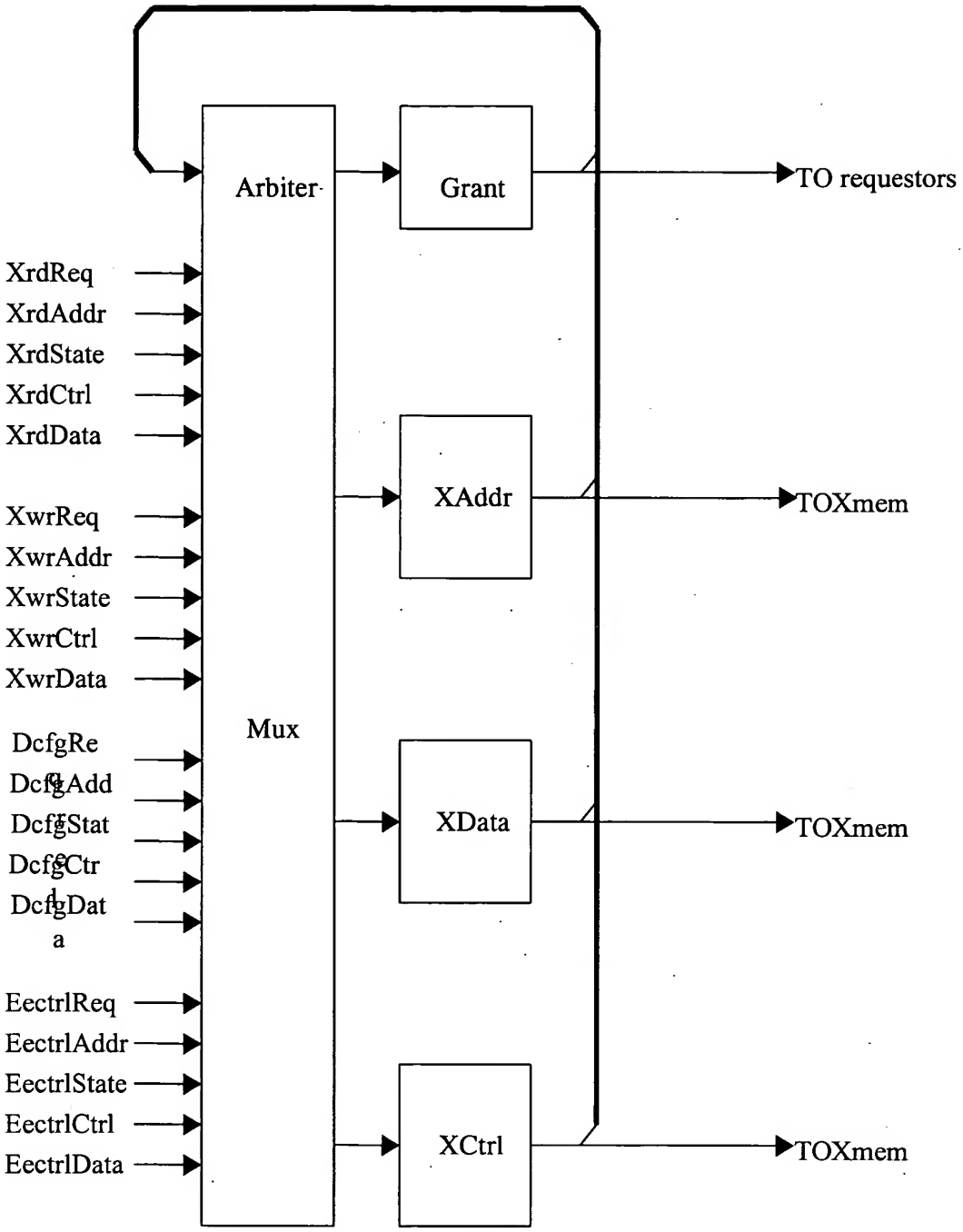
Arbiter
Grant
TO requestors
XrdReq
XrdAddr
XrdState
XrdCtrl
XrdData
XwrReq
XwrAddr
XwrState
XwrCtrl
XwrData
DcfgRe
DcfgAdd
DcfgStat
DcfgCtr
DcfgDat
a
EectrlReq
EectrlAddr
EectrlState
EectrlCtrl
EectrlData
Mux
XAddr
TOXmem
XData
TOXmem
XCtrl
TOXmem

FIG. 28

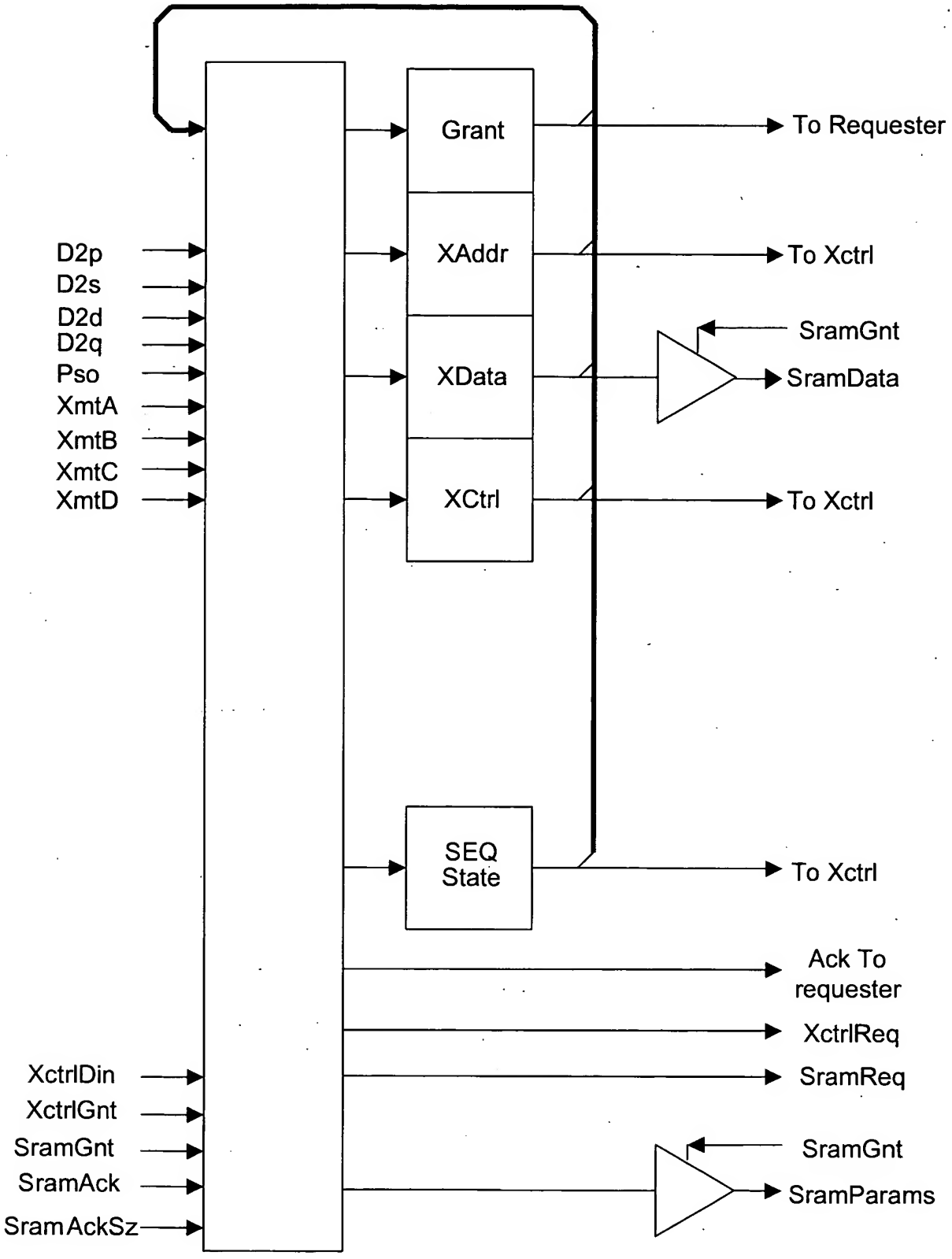
Grant
To Requester
XAddr
To Xctrl
XData
SramGnt
SramData
XCtrl
To Xctrl
SEQ State
To Xctrl
Ack To requester
XctrlReq
SramReq
SramGnt
SramParams
D2p
D2s
D2d
D2q
Pso
XmtA
XmtB
XmtC
XmtD
XctrlDin
XctrlGnt
SramGnt
SramAck
SramAckSz

FIG. 30

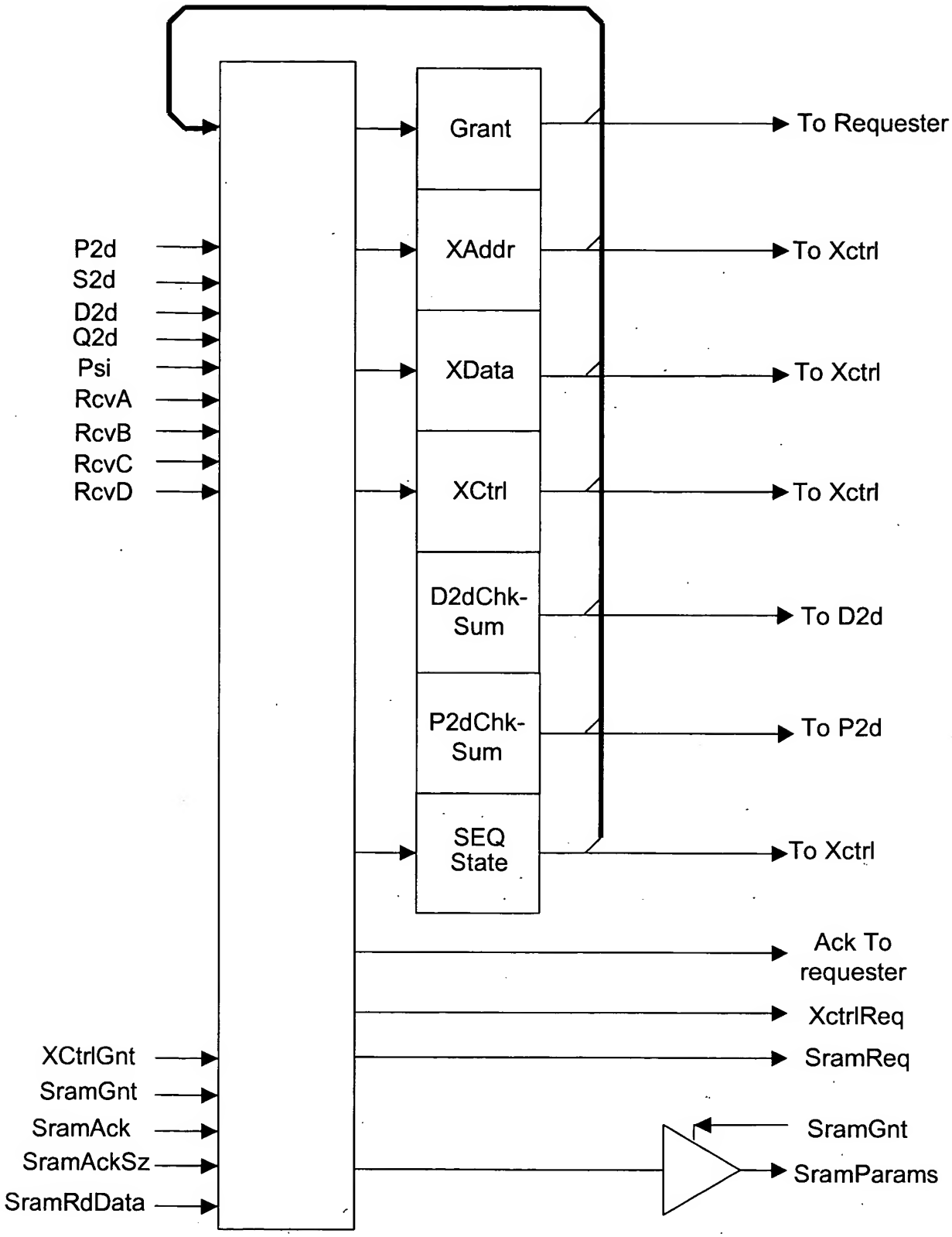
Grant
To Requester
P2d
S2d
D2d
Q2d
Psi
RcvA
RcvB
RcvC
RcvD
XAddr
To Xctrl
XData
To Xctrl
XCtrl
To Xctrl
D2dChk-Sum
To D2d
P2dChk-Sum
To P2d
SEQ State
To Xctrl
Ack To requester
XctrlReq
SramReq
XCtrlGnt
SramGnt
SramAck
SramAckSz
SramRdData
SramGnt
SramParams

FIG. 32

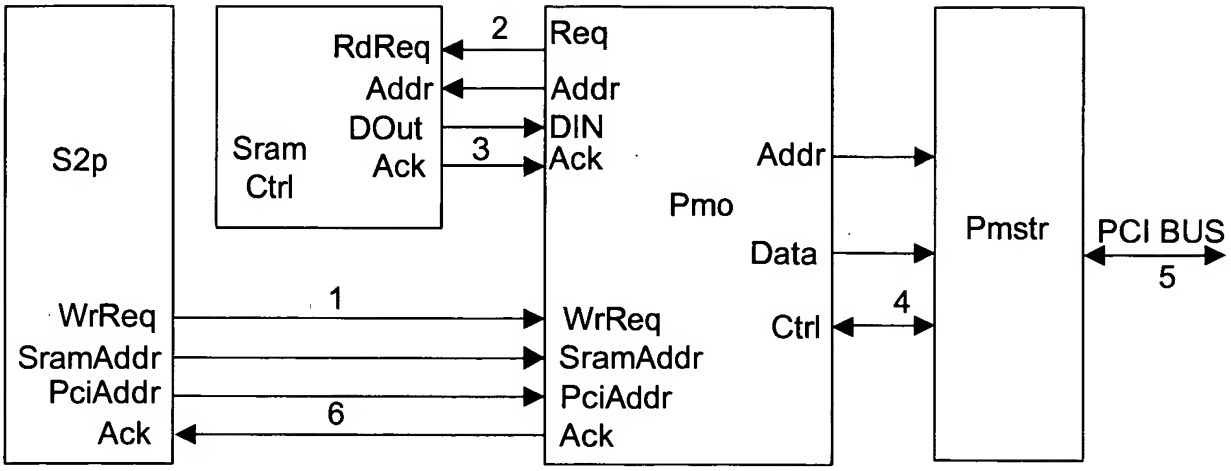
S2p
Sram
Ctrl
RdReq
Addr
DOut
Ack
2
3
Req
Addr
DIN
Ack
Pmo
Addr
Data
Ctrl
4
Pmstr
PCI BUS
5
WrReq
SramAddr
PciAddr
Ack
1
6
WrReq
SramAddr
PciAddr
Ack

FIG. 33

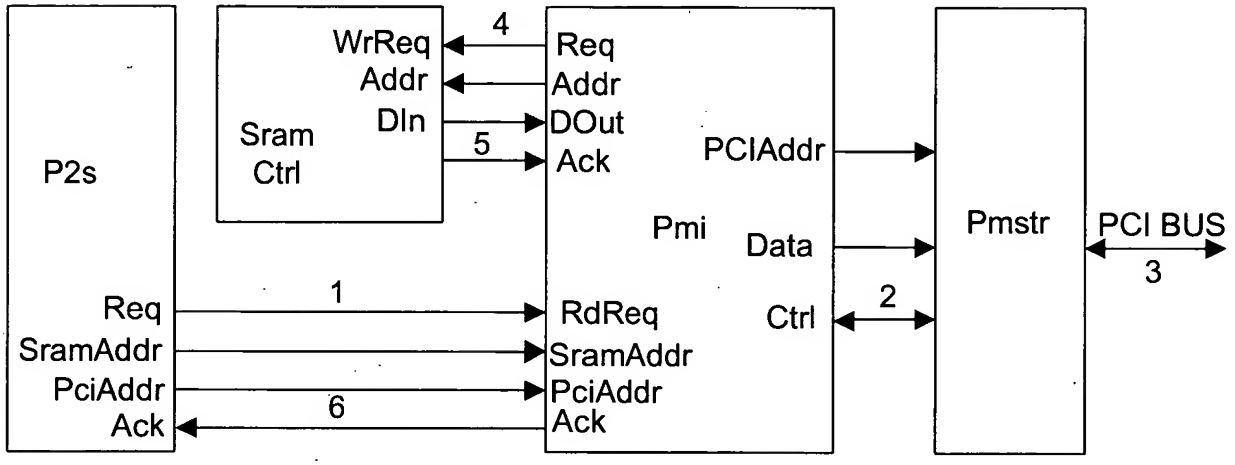
P2s
Sram
Ctrl
WrReq
Addr
DIn
4
5
Req
Addr
DOut
Ack
PCIAddr
Pmi
Data
Ctrl
2
Pmstr
PCI BUS
3
Req
SramAddr
PciAddr
Ack
1
6
RdReq
SramAddr
PciAddr
Ack

FIG. 34

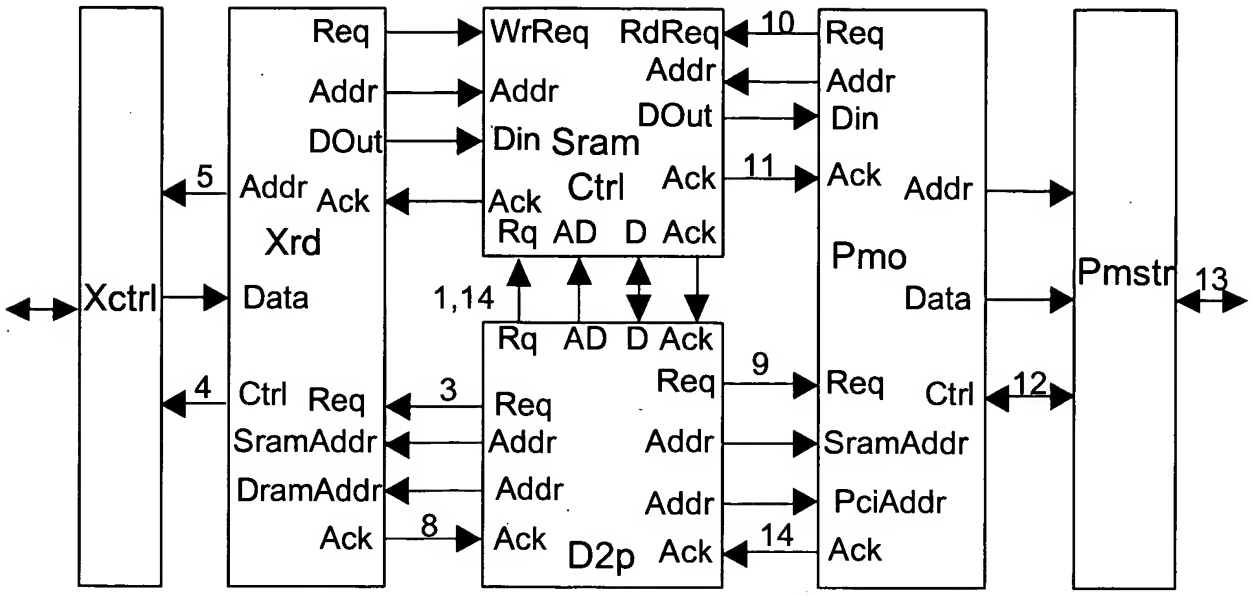
Xctrl
Req
Addr
DOut
5
Addr
Ack
Xrd
Data
4
Ctrl
Req
SramAddr
DramAddr
Ack
WrReq
RdReq
Addr
Addr
Din
DOut
Sram
Ctrl
Ack
Ack
Rq AD D Ack
1,14
Rq AD D Ack
3
Req
Addr
Addr
8
Ack
D2p
10
Req
Addr
Din
11
Ack
Addr
Pmo
Data
9
Req
Ctrl
12
Addr
Addr
PciAddr
14
Ack
Pmstr
13

FIG. 35

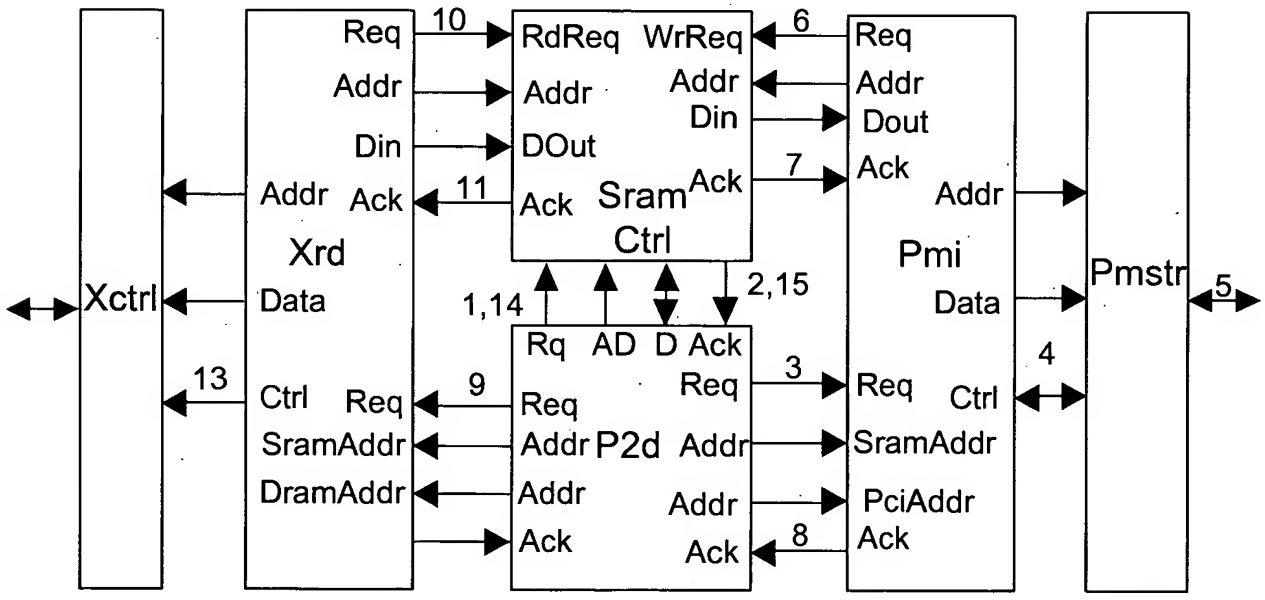
Xctrl
Req
Addr
Din
Addr
Ack
Xrd
Data
13
Ctrl
Req
SramAddr
DramAddr
10
RdReq
WrReq
Addr
Addr
DOut
11
Ack
Sram
Ctrl
Ack
1,14
2,15
Rq AD D Ack
9
Req
Addr
P2d
Addr
Ack
6
Req
Addr
Din
Dout
7
Ack
Addr
Pmi
Data
3
Req
Addr
Addr
8
Ack
Ctrl
SramAddr
PciAddr
Ack
Pmstr
5
4

FIG. 37

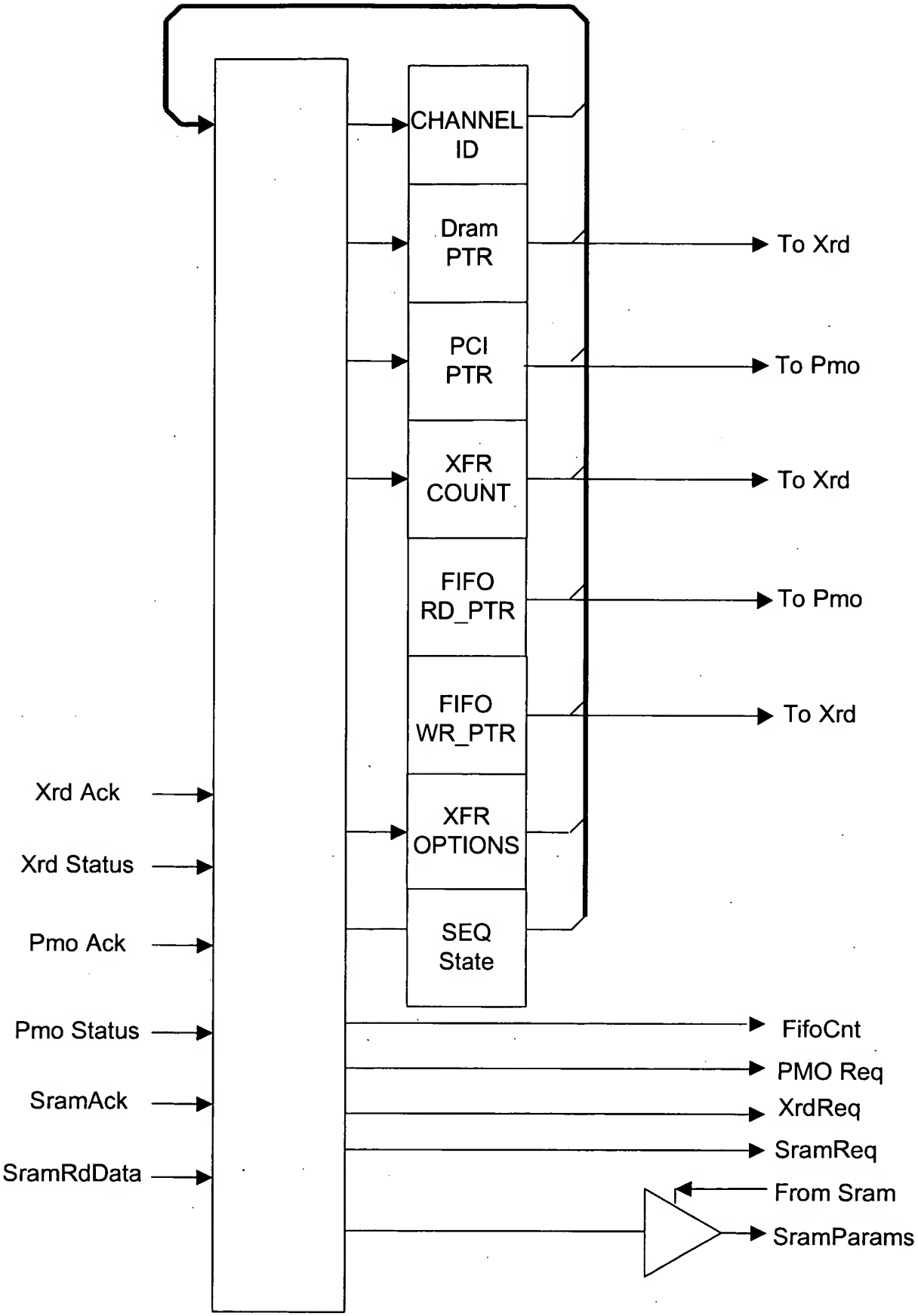
CHANNEL ID
Dram PTR
PCI PTR
XFR COUNT
FIFO RD_PTR
FIFO WR_PTR
XFR OPTIONS
SEQ State
To Xrd
To Pmo
To Xrd
To Pmo
To Xrd
Xrd Ack
Xrd Status
Pmo Ack
Pmo Status
SramAck
SramRdData
FifoCnt
PMO Req
XrdReq
SramReq
From Sram
SramParams

FIG. 36

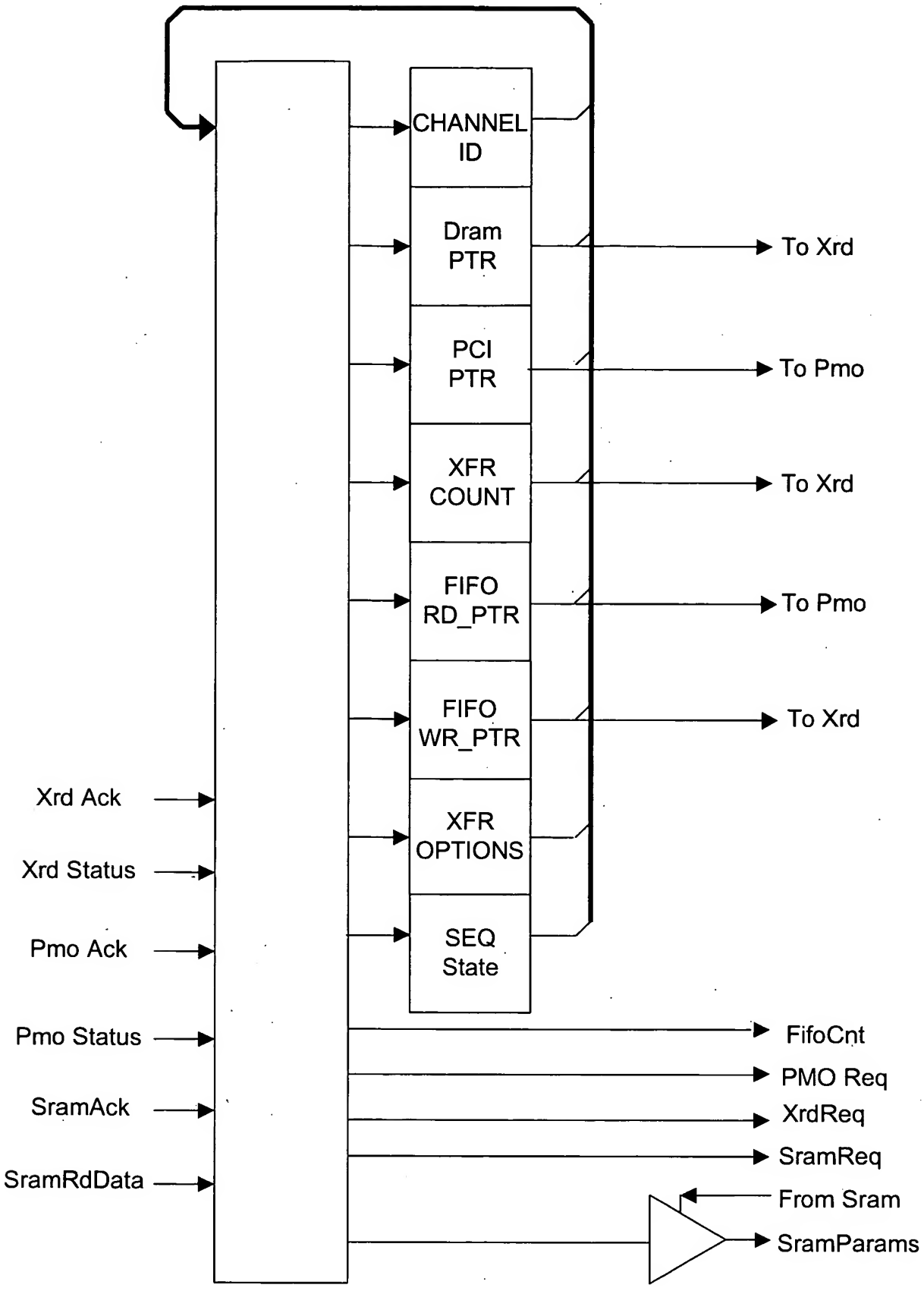
CHANNEL ID
Dram PTR
PCI PTR
XFR COUNT
FIFO RD_PTR
FIFO WR_PTR
XFR OPTIONS
SEQ State
To Xrd
To Pmo
To Xrd
To Pmo
To Xrd
Xrd Ack
Xrd Status
Pmo Ack
Pmo Status
SramAck
SramRdData
FifoCnt
PMO Req
XrdReq
SramReq
From Sram
SramParams

FIG. 38

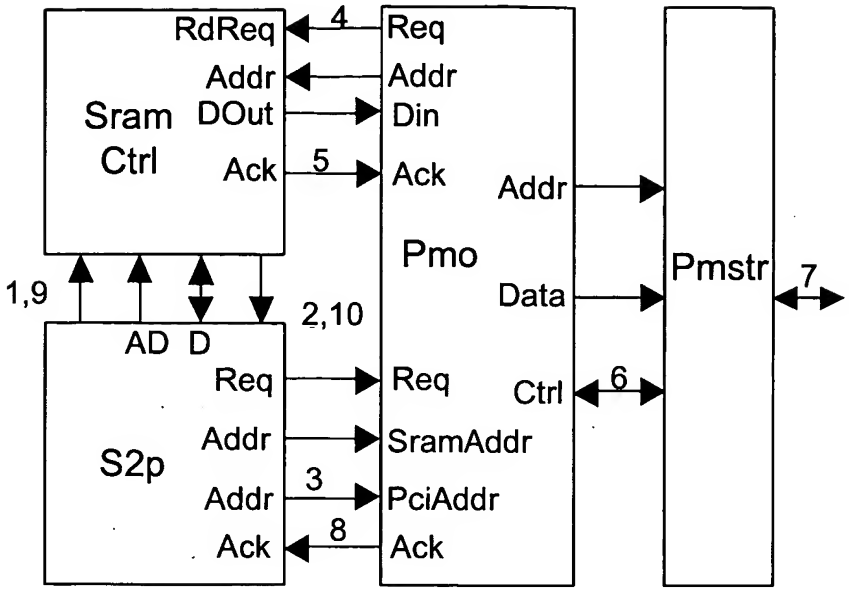
RdReq
Addr
DOut
Ack
Sram
Ctrl
4
5
Req
Addr
Din
Ack
Pmo
Addr
Data
Ctrl
Pmstr
7
6
1,9
2,10
AD
D
Req
Addr
Addr
Ack
S2p
3
8
Req
SramAddr
PciAddr
Ack

FIG. 39

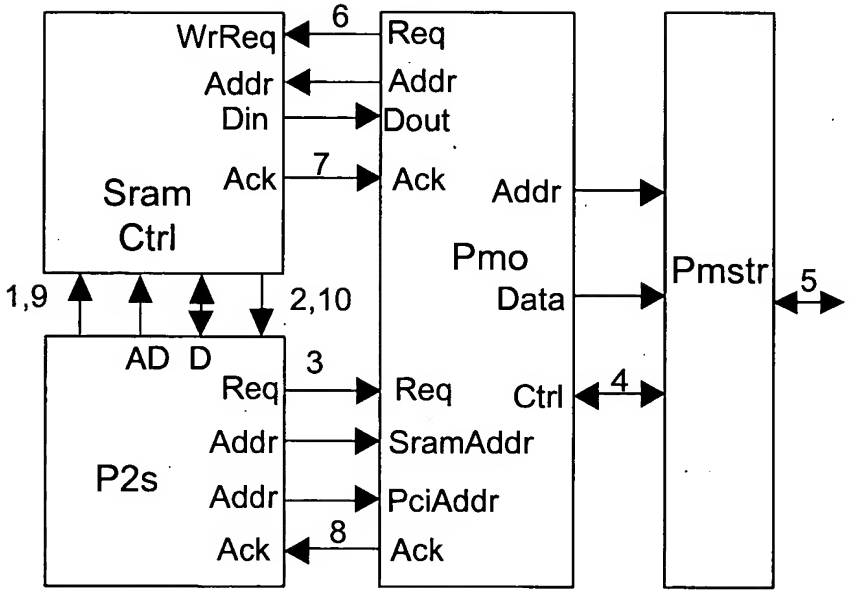
WrReq
Addr
Din
Ack
Sram
Ctrl
6
7
Req
Addr
Dout
Ack
Pmo
Addr
Data
Ctrl
Pmstr
5
4
1,9
2,10
AD
D
Req
Addr
Addr
Ack
P2s
3
8
Req
SramAddr
PciAddr
Ack

FIG. 41

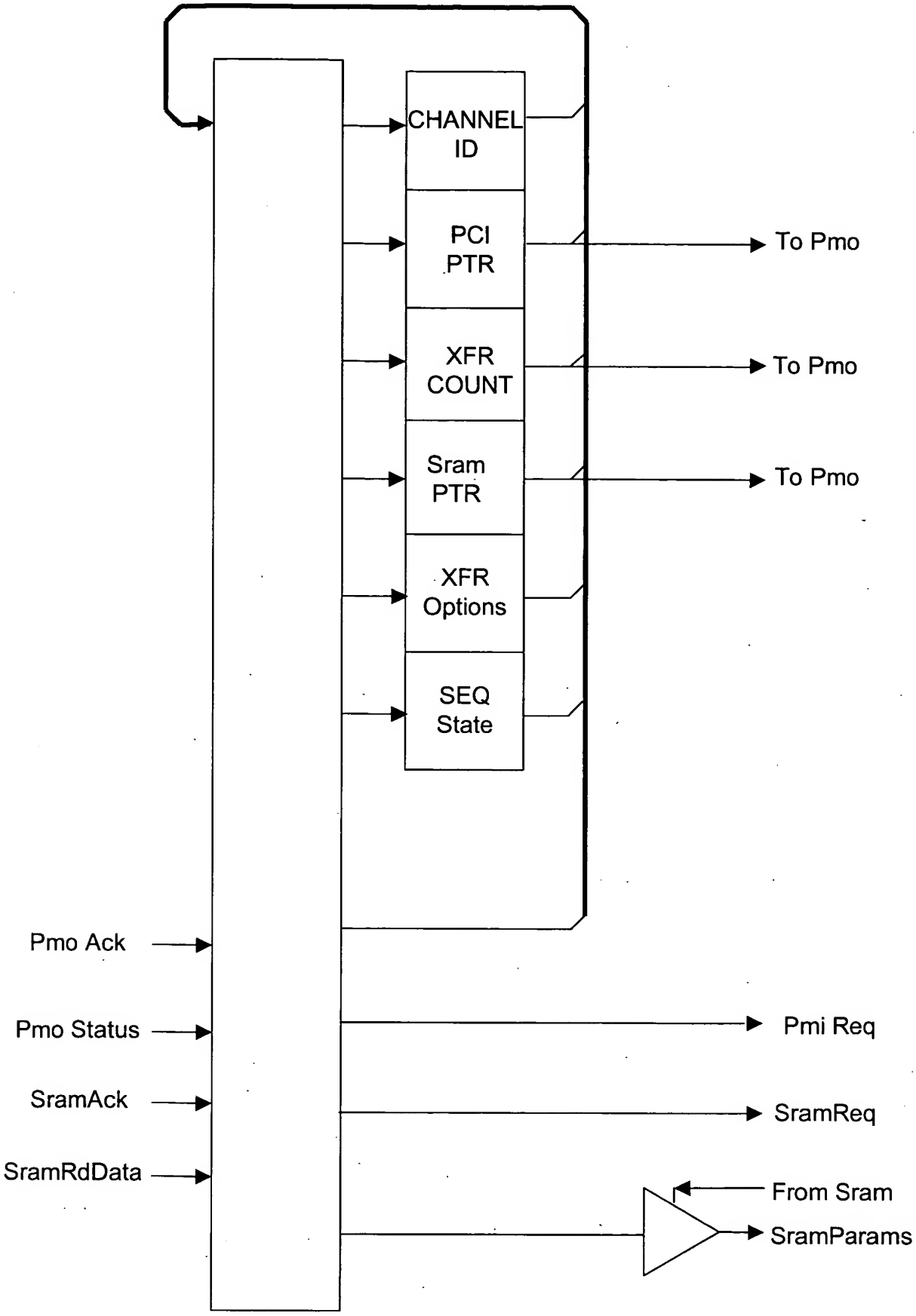
CHANNEL ID
PCI PTR
XFR COUNT
Sram PTR
XFR Options
SEQ State
To Pmo
To Pmo
To Pmo
Pmo Ack
Pmo Status
SramAck
SramRdData
Pmi Req
SramReq
From Sram
SramParams

FIG. 40

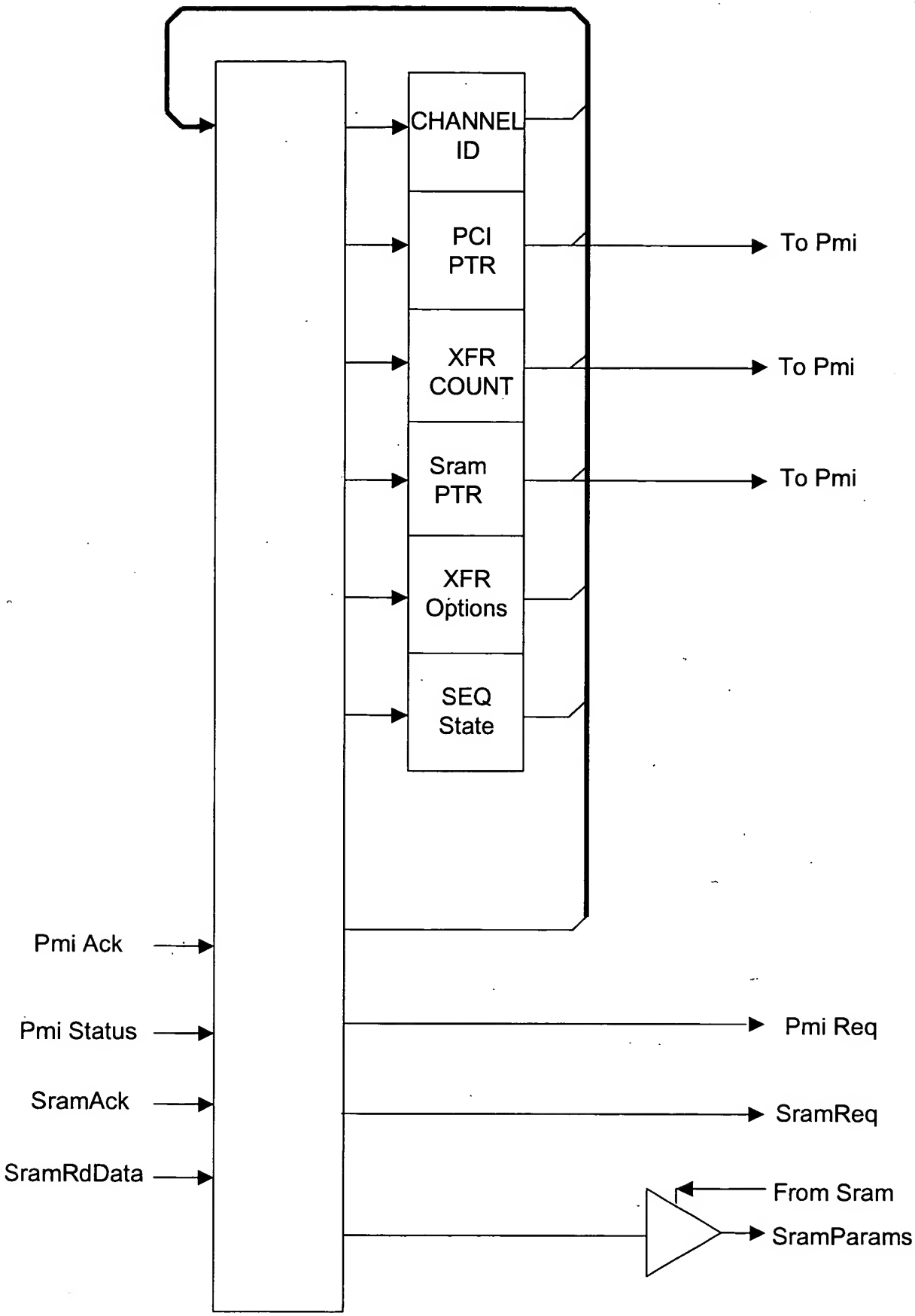
CHANNEL ID
PCI PTR
XFR COUNT
Sram PTR
XFR Options
SEQ State
To Pmi
To Pmi
To Pmi
Pmi Ack
Pmi Status
SramAck
SramRdData
Pmi Req
SramReq
From Sram
SramParams

FIG. 42

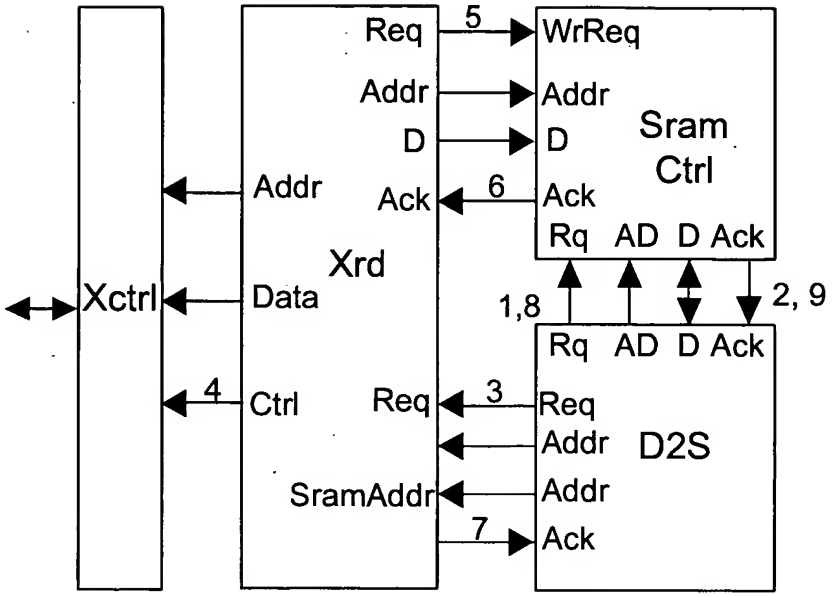
Req
5
WrReq
Addr
Addr
D
D
Sram
Ctrl
Addr
Ack
6
Ack
Rq AD D Ack
Xrd
Xctrl
Data
1,8
2, 9
Rq AD D Ack
4
Ctrl
Req
3
Req
Addr
D2S
SramAddr
Addr
7
Ack

FIG. 43

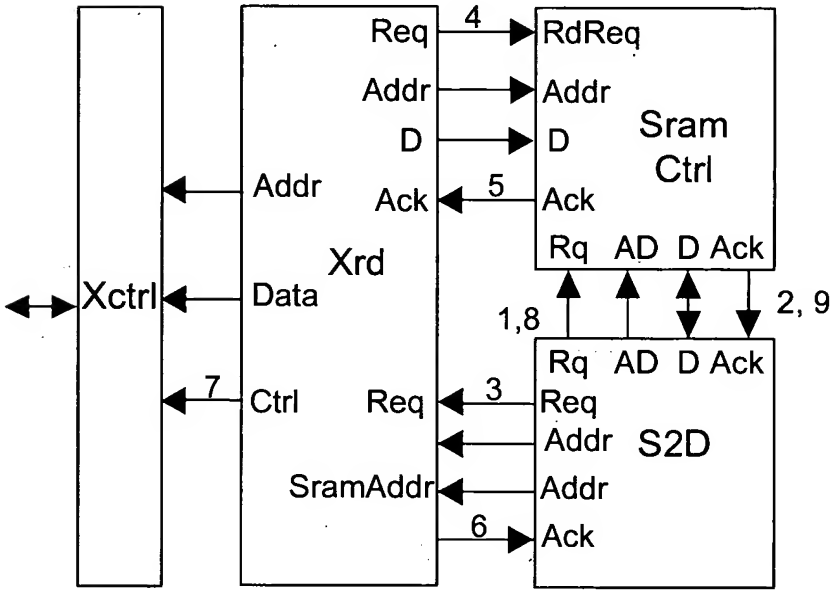
Req
4
RdReq
Addr
Addr
D
D
Sram
Ctrl
Addr
Ack
5
Ack
Rq AD D Ack
Xrd
Xctrl
Data
1,8
2, 9
Rq AD D Ack
7
Ctrl
Req
3
Req
Addr
S2D
SramAddr
Addr
6
Ack

FIG. 45

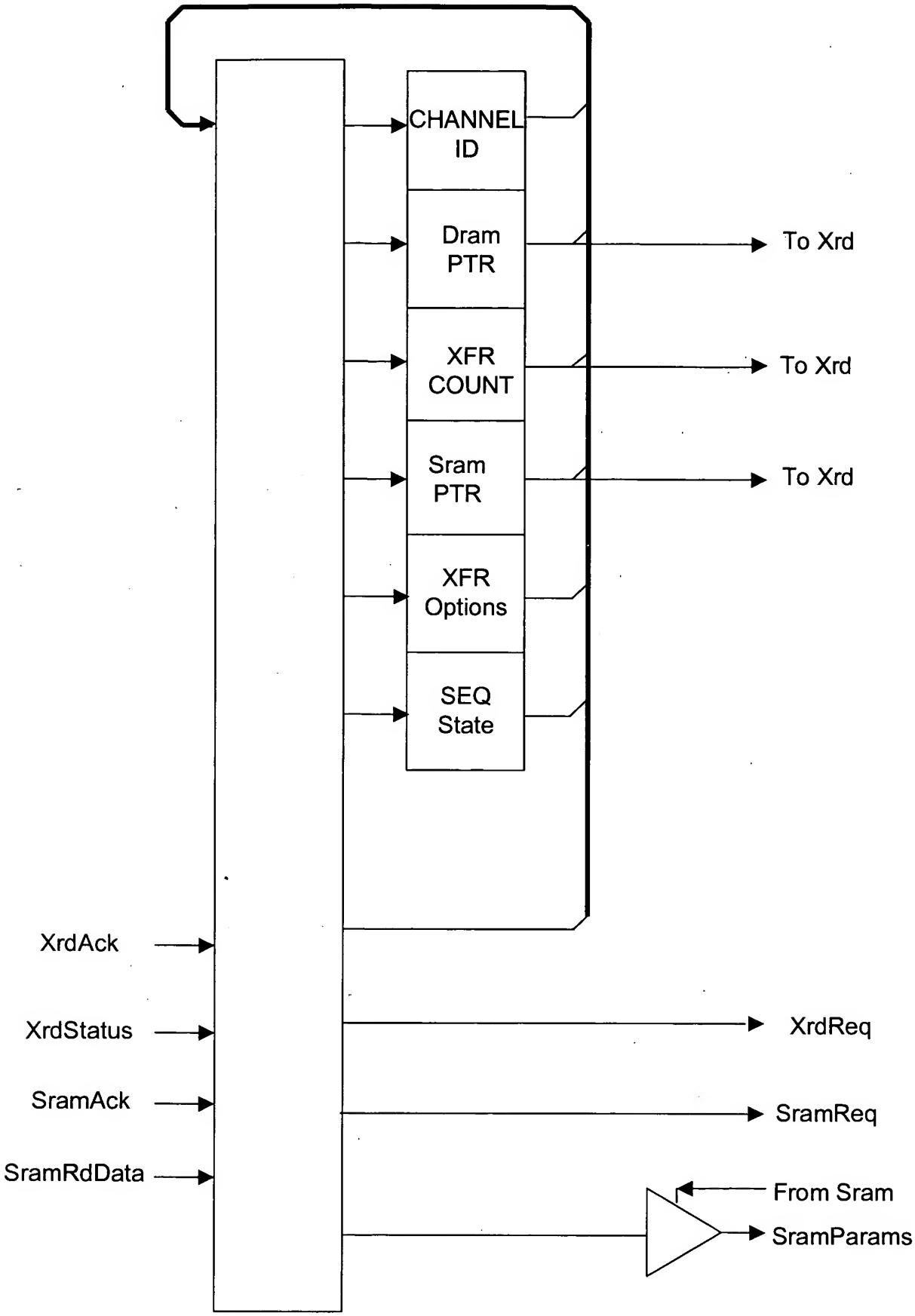
CHANNEL ID
Dram PTR
XFR COUNT
Sram PTR
XFR Options
SEQ State
To Xrd
To Xrd
To Xrd
XrdAck
XrdStatus
SramAck
SramRdData
XrdReq
SramReq
From Sram
SramParams

FIG. 44

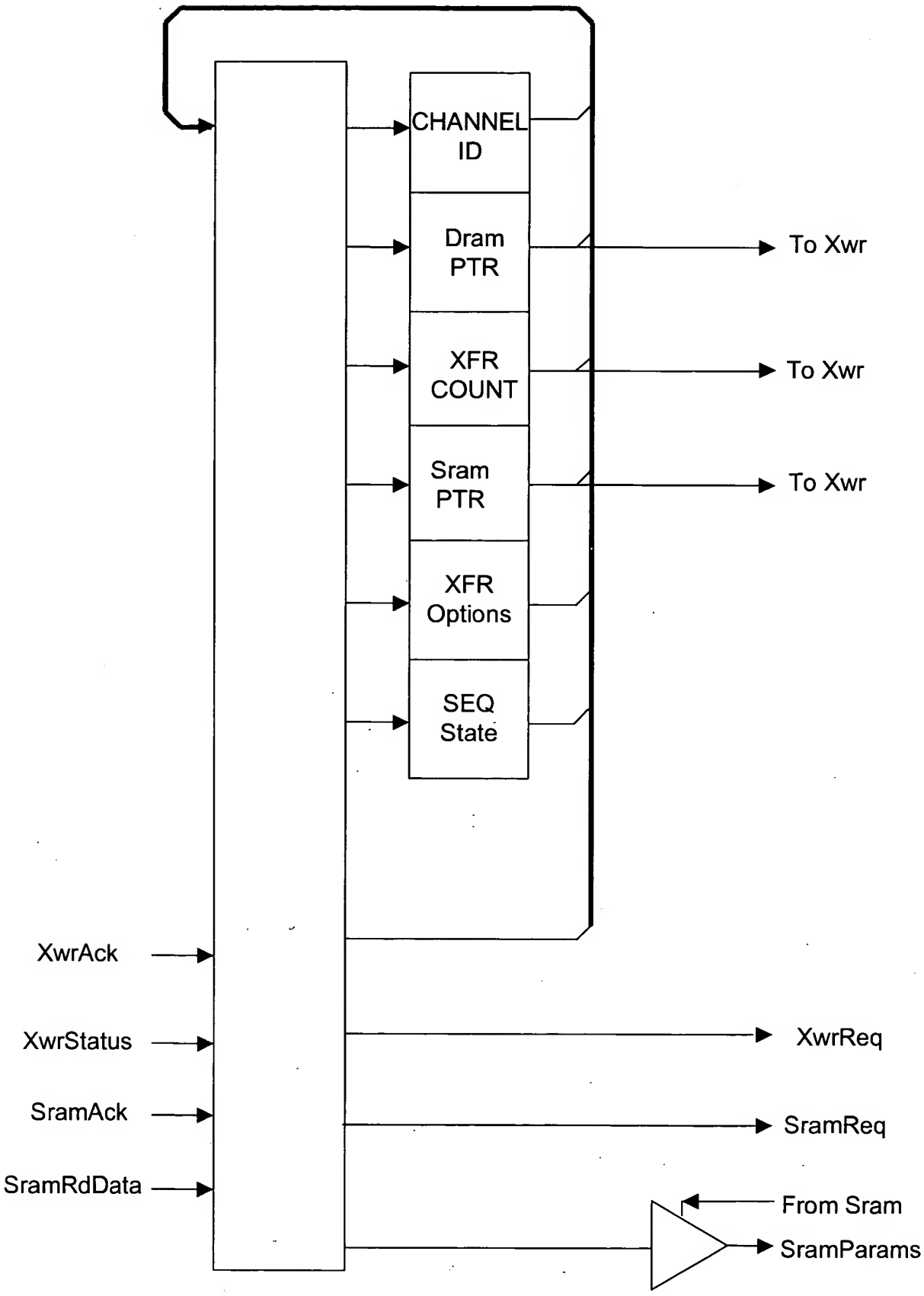
CHANNEL ID
Dram PTR
XFR COUNT
Sram PTR
XFR Options
SEQ State
To Xwr
To Xwr
To Xwr
XwrAck
XwrStatus
SramAck
SramRdData
XwrReq
SramReq
From Sram
SramParams

FIG. 46

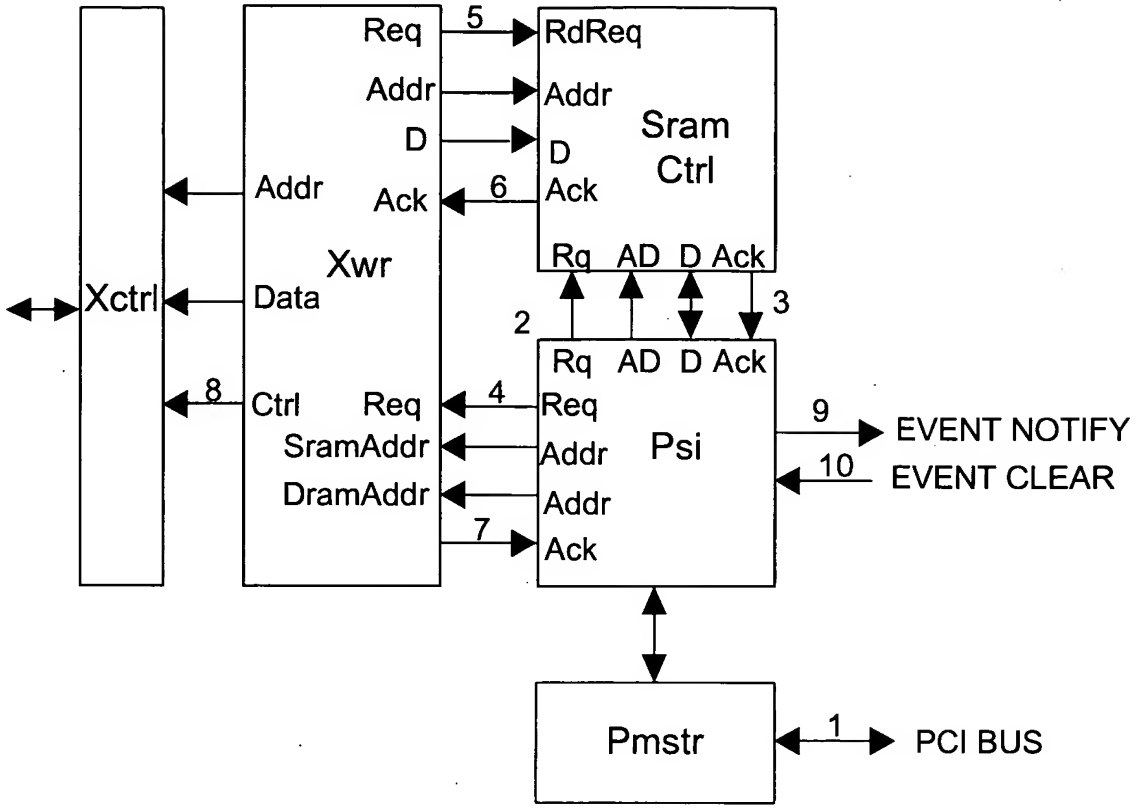
Req
Addr
D
Addr
Ack
Xwr
Data
Ctrl
Req
SramAddr
DramAddr
Xctrl
5
6
8
4
7
RdReq
Addr
D
Ack
Sram
Ctrl
Rq AD D Ack
2
3
Rq AD D Ack
Req
Addr
Addr
Ack
Psi
9
EVENT NOTIFY
10
EVENT CLEAR
Pmstr
1
PCI BUS

FIG. 47

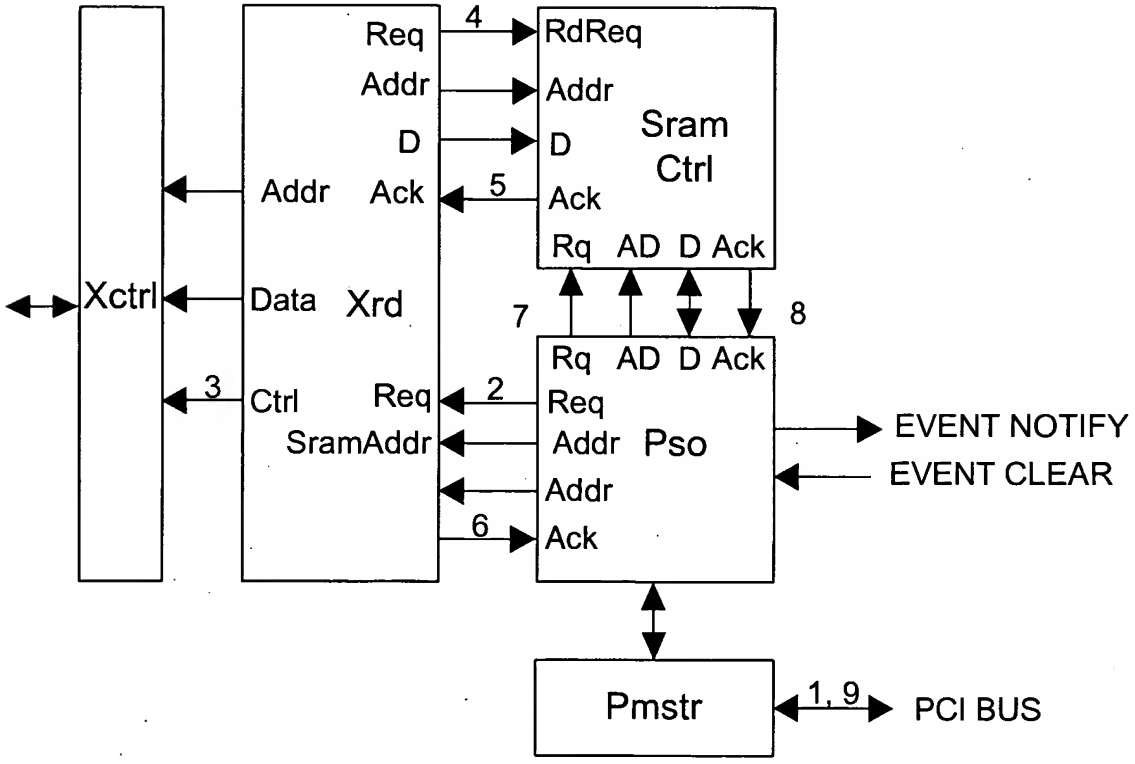
Req
Addr
D
Addr
Ack
Xrd
Data
Ctrl
Req
SramAddr
Xctrl
4
5
3
2
6
RdReq
Addr
D
Ack
Sram
Ctrl
Rq AD D Ack
7
8
Rq AD D Ack
Req
Addr
Addr
Ack
Pso
EVENT NOTIFY
EVENT CLEAR
Pmstr
1, 9
PCI BUS

FIG. 48

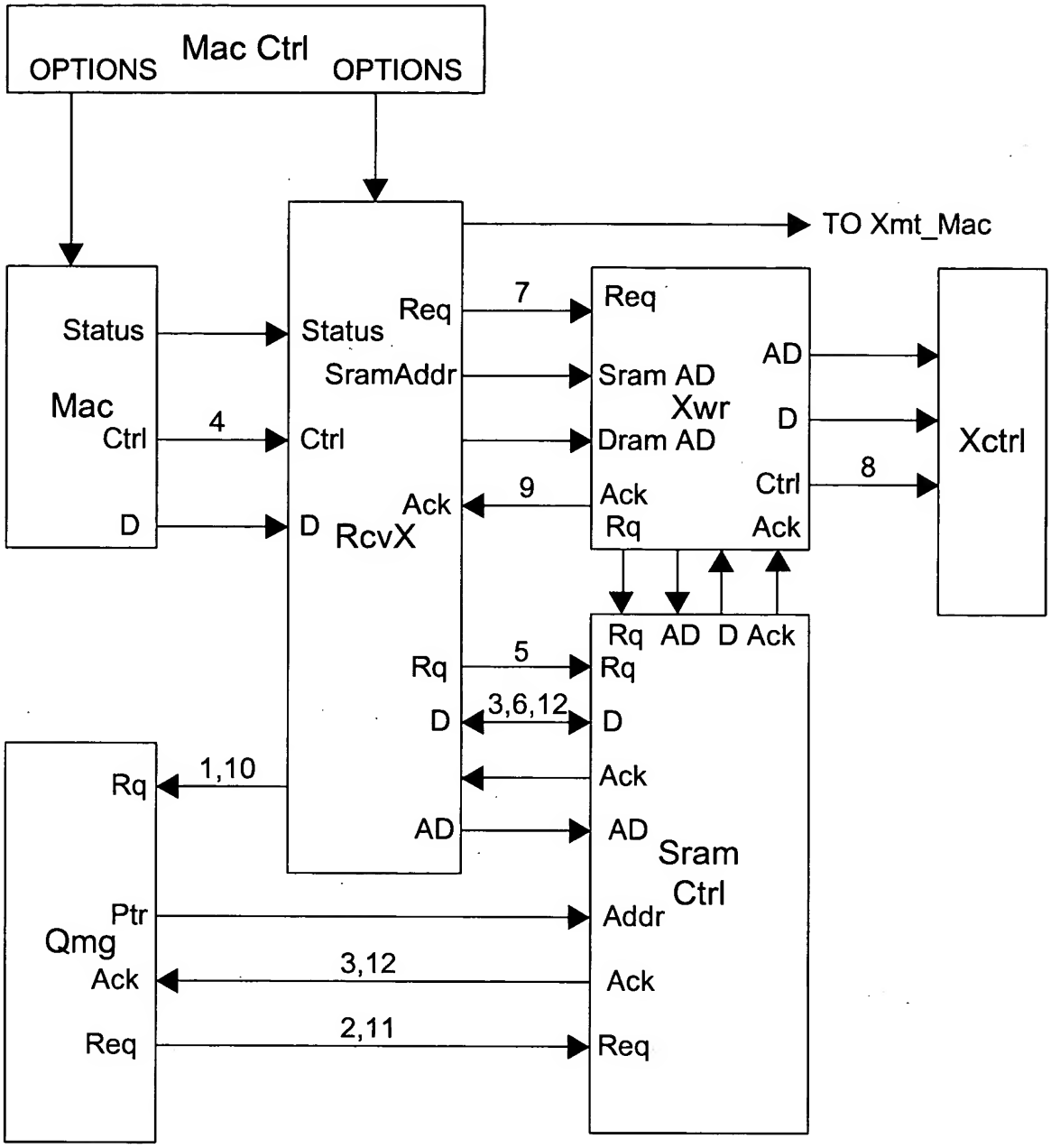
Mac Ctrl
OPTIONS
OPTIONS
TO Xmt_Mac
Status
Mac
Ctrl
D
4
Status
Req
SramAddr
Ctrl
D
RcvX
Ack
9
7
Req
Sram AD
Xwr
Dram AD
Ack
Rq
AD
D
Ctrl
8
Ack
Xctrl
Rq AD D Ack
Rq
5
D
3,6,12
Ack
AD
Sram
Ctrl
Addr
Ack
Req
Rq
1,10
Ptr
Qmg
Ack
3,12
Req
2,11

FIG. 49

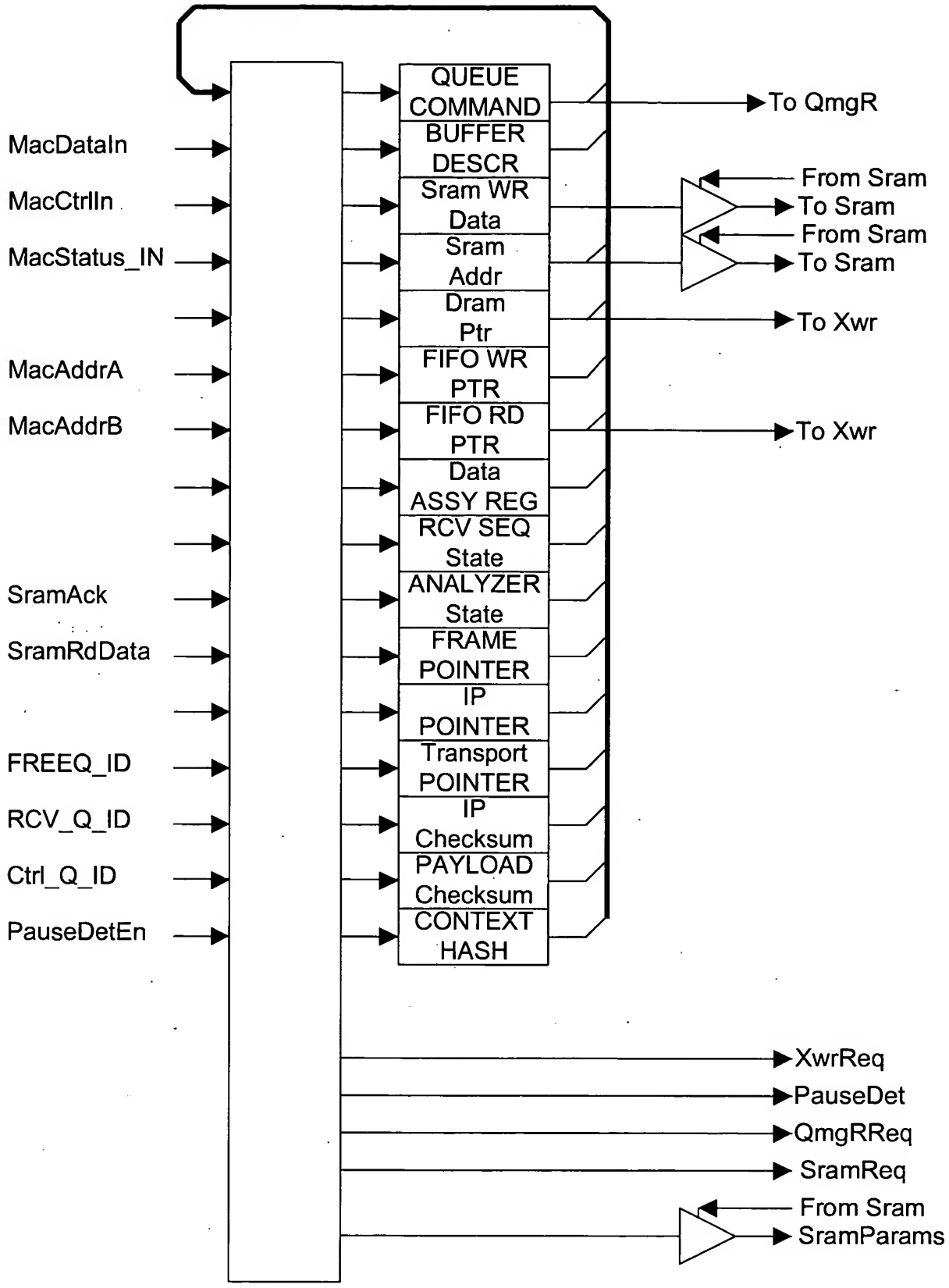
QUEUE COMMAND
BUFFER DESCR
Sram WR Data
Sram Addr
Dram Ptr
FIFO WR PTR
FIFO RD PTR
Data ASSY REG
RCV SEQ State
ANALYZER State
FRAME POINTER
IP POINTER
Transport POINTER
IP Checksum
PAYLOAD Checksum
CONTEXT HASH
MacDataIn
MacCtrlIn
MacStatus_IN
MacAddrA
MacAddrB
SramAck
SramRdData
FREEQ_ID
RCV_Q_ID
Ctrl_Q_ID
PauseDetEn
To QmgR
From Sram
To Sram
From Sram
To Sram
To Xwr
To Xwr
XwrReq
PauseDet
QmgRReq
SramReq
From Sram
SramParams

FIG. 50

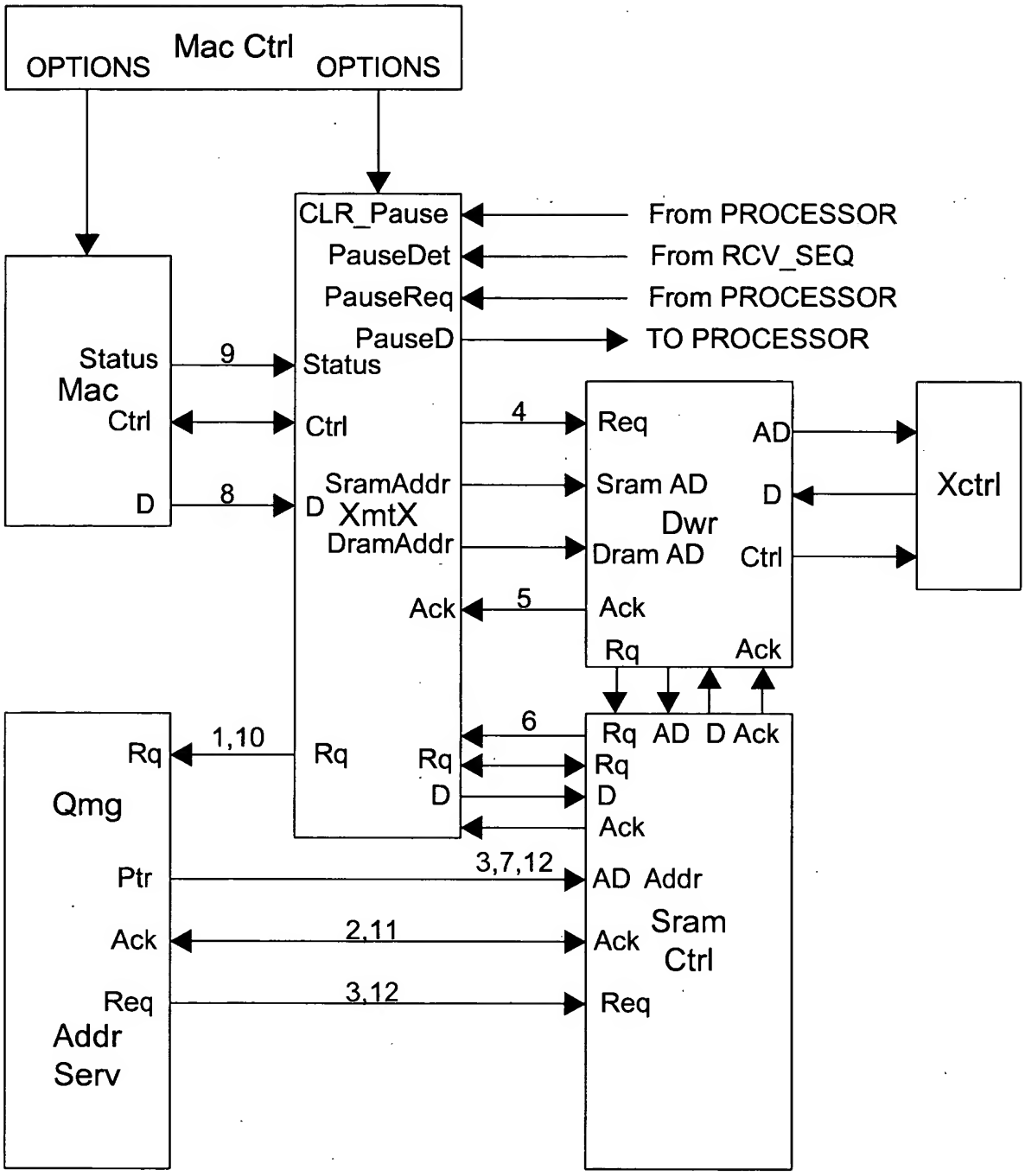
Mac Ctrl
OPTIONS
OPTIONS
CLR_Pause
From PROCESSOR
PauseDet
From RCV_SEQ
PauseReq
From PROCESSOR
PauseD
TO PROCESSOR
Status
Mac
Ctrl
D
9
8
Status
Ctrl
SramAddr
D
XmtX
DramAddr
Ack
4
5
Req
Sram AD
Dwr
Dram AD
Ack
Rq
AD
D
Ctrl
Ack
Xctrl
Rq
Ack
1,10
Rq
Rq
D
6
Rq AD D Ack
Rq
D
Ack
Qmg
Ptr
Ack
Req
Addr
Serv
3,7,12
2,11
3,12
AD Addr
Ack
Sram
Ctrl
Req

FIG. 51

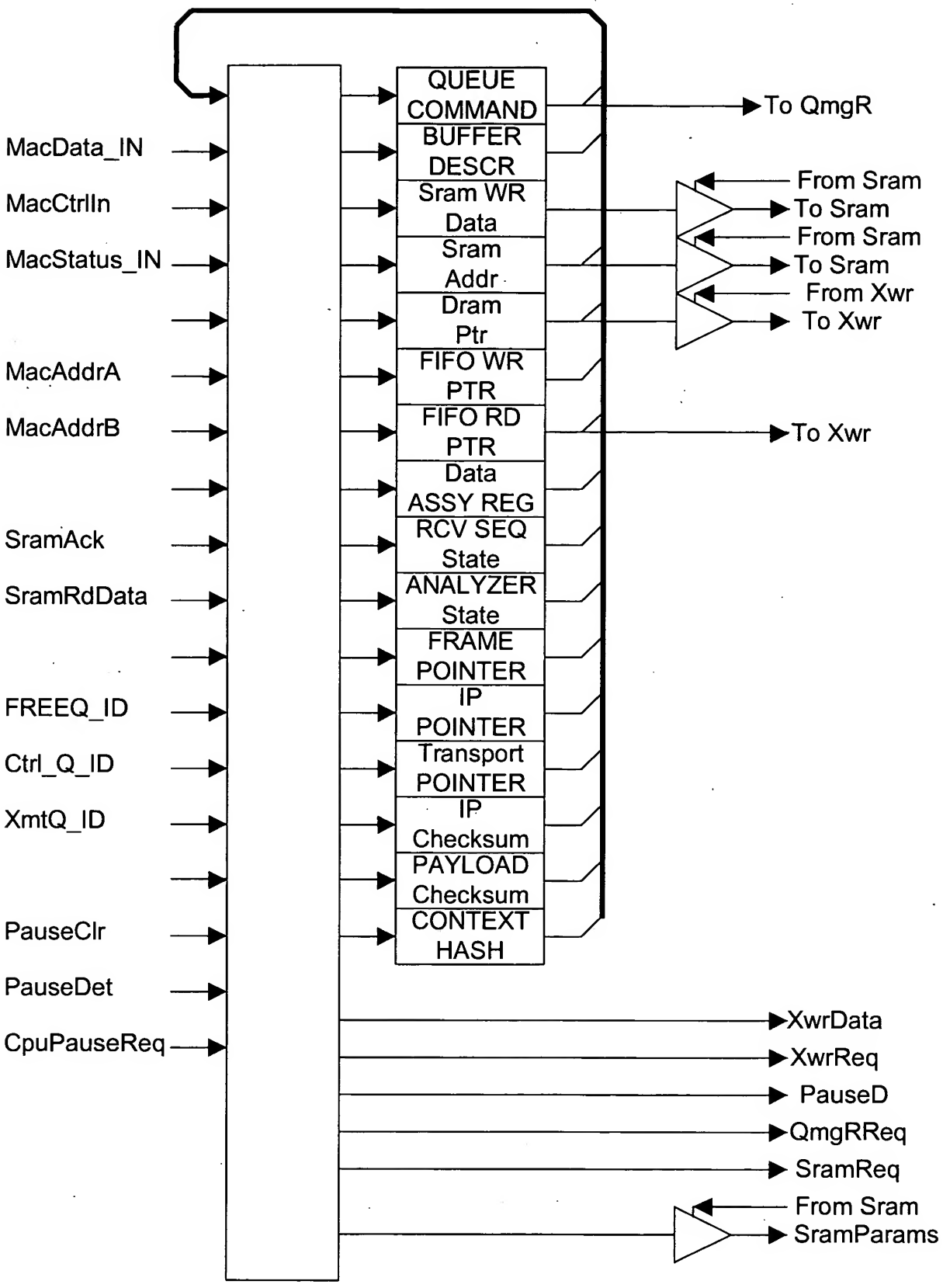
QUEUE COMMAND
BUFFER DESCR
Sram WR Data
Sram Addr
Dram Ptr
FIFO WR PTR
FIFO RD PTR
Data ASSY REG
RCV SEQ State
ANALYZER State
FRAME POINTER
IP POINTER
Transport POINTER
IP Checksum
PAYLOAD Checksum
CONTEXT HASH
To QmgR
From Sram
To Sram
From Sram
To Sram
From Xwr
To Xwr
To Xwr
MacData_IN
MacCtrlIn
MacStatus_IN
MacAddrA
MacAddrB
SramAck
SramRdData
FREEQ_ID
Ctrl_Q_ID
XmtQ_ID
PauseClr
PauseDet
CpuPauseReq
XwrData
XwrReq
PauseD
QmgRReq
SramReq
From Sram
SramParams

FIG. 52

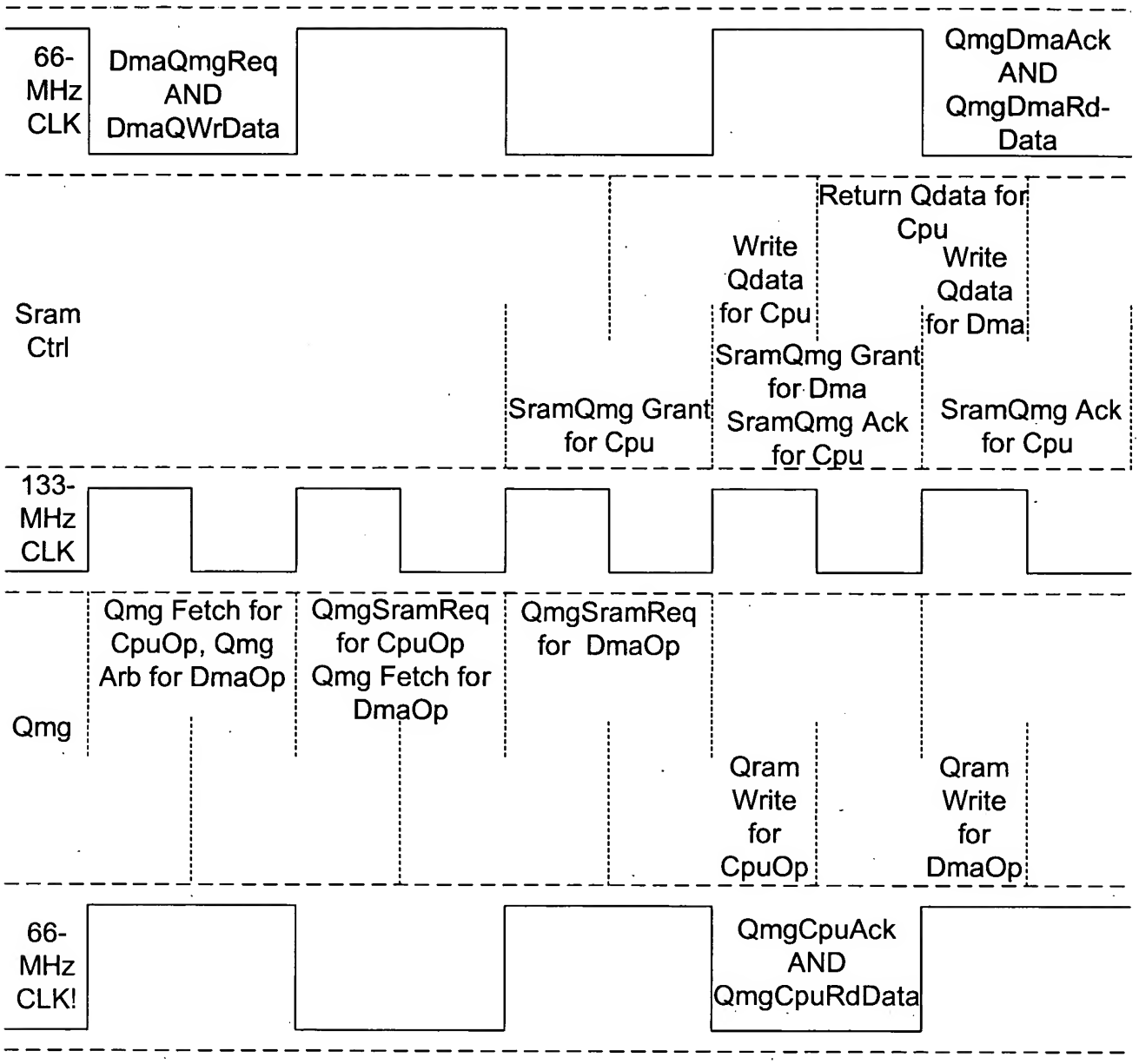
66-MHz CLK
DmaQmgReq AND DmaQWrData
QmgDmaAck AND QmgDmaRd-Data
Sram Ctrl
SramQmg Grant for Cpu
Write Qdata for Cpu
Return Qdata for Cpu
SramQmg Grant for Dma
SramQmg Ack for Cpu
Write Qdata for Dma
SramQmg Ack for Cpu
133-MHz CLK
Qmg
Qmg Fetch for CpuOp, Qmg Arb for DmaOp
QmgSramReq for CpuOp Qmg Fetch for DmaOp
QmgSramReq for DmaOp
Qram Write for CpuOp
Qram Write for DmaOp
66-MHz CLK!
QmgCpuAck AND QmgCpuRdData

FIG. 53

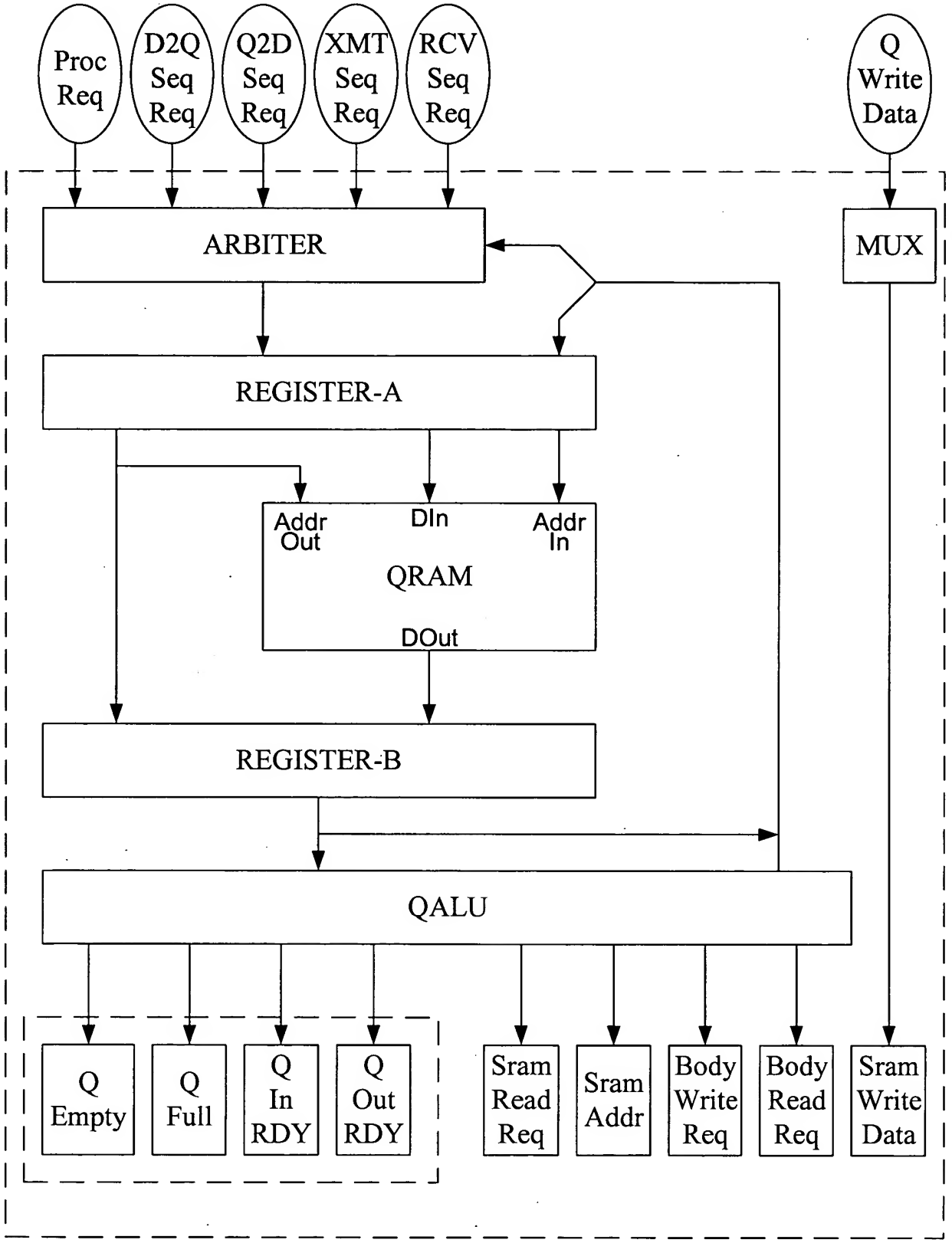
Proc Req
D2Q Seq Req
Q2D Seq Req
XMT Seq Req
RCV Seq Req
Q Write Data
ARBITER
MUX
REGISTER-A
Addr Out
DIn
Addr In
QRAM
DOut
REGISTER-B
QALU
Q Empty
Q Full
Q In RDY
Q Out RDY
Sram Read Req
Sram Addr
Body Write Req
Body Read Req
Sram Write Data

FIG. 54